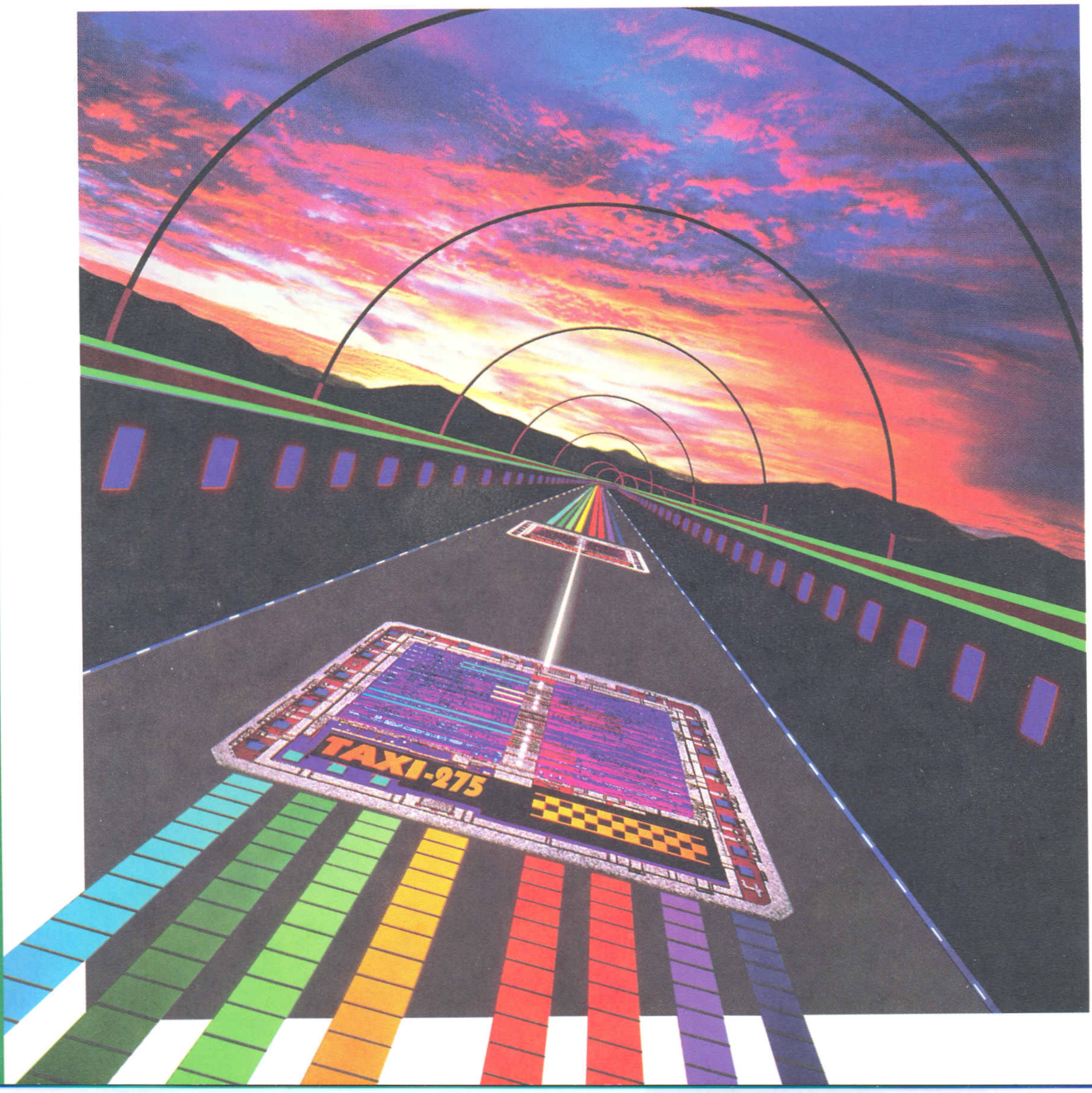
Am79168/Am79169 TAXI™-275

Technical Manual

Advanced
Micro
Devices



Am79168/Am79169-275 TAXI-275 Integrated Circuits

Technical Manual

Rev. 1.0, 1993

ADVANCED MICRO DEVICES

Advanced Micro Devices reserves the right to make changes in its products without notice in order to improve design or performance characteristics.

This publication neither states nor implies any warranty of any kind, including but not limited to implied warrants of merchantability or fitness for a particular application. AMD® assumes no responsibility for the use of any circuitry other than the circuitry in an AMD product.

The information in this publication is believed to be accurate in all respects at the time of publication, but is subject to change without notice. AMD assumes no responsibility for any errors or omissions, and disclaims responsibility for any consequences resulting from the use of the information included herein. Additionally, AMD assumes no responsibility for the functioning of undescribed features or parameters.

Trademarks

TAXI and TAXIchip are trademarks of Advanced Micro Devices, Inc.

Product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

TABLE OF CONTENTS

Chapter 1.0 **TAXI-275 INTRODUCTION**

1.1 Transmitter Overview 1-1
1.2 Receiver Overview 1-2
1.3 Using the TAXI-275 Chipset 1-3
1.4 Data Widths, Data and Commands 1-4
1.5 Operational Modes—Normal, Raw, Loopback, SYNCD, Test 1-5

Chapter 2.0 **DATA ENCODING OVERVIEW**

2.1 Unencoded (Raw) Data Transmission 2-2
2.2 Violation Detection 2-2
2.3 Receiver Byte Framing (Use of SYNC Character) 2-3

Chapter 3.0 **TIMING CONSIDERATIONS**

3.1 Phased-Locked Loop 3-1
3.1.1 Introduction 3-1
3.1.2 Data Acquisition Time (quiet to lock time) 3-1
3.1.3 Data Acquisition Time (power-up to lock time) 3-2
3.2 Asynchronous TAXI-275 Interfaces (STRBI/ACK) 3-3
3.3 Synchronous TAXI-275 Interface (STRBI/CLKI) 3-4
3.4 TX/RX Latency 3-4

Chapter 4.0 **INTERFACE TO SERIAL MEDIA**

4.1 Coaxial and Fiber Optic Module Interface Decoupling 4-1
4.1.1 Introduction 4-1
4.1.2 Very Short Links 4-1
4.1.3 Terminated, DC Coupled 4-2
4.1.4 Terminated, AC Capacitor Coupled 4-2
4.1.5 Baseline Wander and AC Capacitor Coupling 4-4
4.1.6 Interfacing to Fiber Optic Transmitters/Receivers 4-5
4.1.7 DC Coupled TAXI-275 Fiber Optic Interface 4-5
4.1.8 AC Coupled TAXI Optical Fiber Interface 4-6
4.1.9 Interfacing to Coaxial Cable 4-7
4.2 Interfacing to Twisted-Pair Cable 4-8
4.3 AC Transformer Coupled 4-9
4.4 Compensation 4-10
4.5 Link Performance 4-11
4.5.1 Jitter 4-11
4.5.2 Distance/Failure Knee 4-11
4.6 Cable Disconnect Detection Design 4-12

Chapter 5.0 **BOARD LAYOUT CONSIDERATIONS**

5.1 Printed Circuit Board Layout 5-1
5.1.1 Rules for Layout 5-1
5.2 Layout using Fiber Optic Data Links 5-3

Chapter 6.0 INTERFACING TO WIDE BUSSES

6.1 Wide Bus Interfacing with the Transmitter 6-1
6.1.1 Introduction 6-1
6.1.2 Advantages 6-1
6.1.3 Implementation 6-1
6.1.4 Operation 6-2
6.2 Demultiplexing a TAXI-275 Receiver to N-Byte Parallel Output 6-4
6.2.1 Introduction 6-4
6.2.2 Functional Description For Example Design 6-5
6.2.3 Timing Considerations 6-9

Chapter 7.0 UPGRADING FROM Am7968/69 TO Am79168/169-275

7.1 Introduction 7-1
7.2 Basic Features of The Am79168/9-275 Devices 7-1
7.3 Specific Functionality Differences 7-2
7.3.1 Transmitter—Am7968 vs. Am79168-275 7-2
7.3.2 Receiver—Am7969 vs. Am79169-275 7-4

Appendix A TEST MODE OPERATION

A.1 Introduction A-1
A.2 Transmitter Connections A-1
A.3 Timing Relationships in Test Mode A-2

Appendix B APPLICATION DESIGN NOTES

B.1 Introduction to the Emerging High Speed Data Communication Standards B-1
B.2 The TAXI-275 and ESCON/Fibre Channel B-1
B.3 Asynchronous Transfer Mode (ATM) B-2

Appendix C VENDOR AND MANUFACTURING LISTING

C.1 Fiber Optic Media C-1
C.2 Copper Media C-2
C.3 Crystal Oscillators C-2
C.4 Capacitors, Ferrite Beads, Transformers, misc C-2

Appendix D DATA ENCODING

D.1 ANSI Fibre Channel 8B10B Transmission Block Code D-1
D.1.1 Introduction to 8B10B Code D-1
D.1.2 Disparity and Running Disparity D-1
D.1.3 Principles D-2
D.1.4 A Simple Example of 8B10B Encoding D-13
D.1.5 Special Characters D-13
D.1.6 Exceptions to Encode/Decode Rules D-14
D.1.7 Error Detection D-15
D.1.8 Summary D-17
D.1.9 References D-17
D.2 10B12B Transmission Block Code D-18
D.2.1 An Example of 10B12B Encoding D-19
D.2.2 Special Characters D-19

Appendix E INTRODUCTION TO HIGH SPEED COMMUNICATION STANDARDS

E.1 Introduction to ESCON E-1

E.1.1 The Old Copper Channel versus ESCON E-1

E.1.2 ESCON Physical Links E-3

E.1.3 ESCON Link-Level Functions E-3

E.1.4 ESCON Device-Level Functions E-4

E.2 ANSI Fibre Channel Standard (FCS) E-5

E.2.1 Introduction to FCS E-5

E.2.2 Why do we need Fibre Channel? E-6

E.2.3 FCS Physical Links E-6

E.2.3.1 Laser Safety E-7

E.2.3.2 FCS Transmission Block Code and FC-1 . . . . E-8

E.2.4 Fibre Channel Frame Format E-9

E.2.5 FCS Link-Level Functions E-9

E.2.6 FCS Fabric E-12

E.3 ESCON versus Fibre Channel E-12

LIST OF TABLES

Table 4.1 Copper Media Test Results 4-12

Table 7.1 Transmitter—Am7968 vs. Am79168-275 7-2

Table 7.2 Receiver—Am7969 vs. Am79169-275 7-4

Table D.1 8B10B Encode/Decode Table D-3

Table D.2 Complete 8B10B Encoding Table D-5

Table D.3 10B12B Encode/Decode Table D-18

Table E.1 ESCON Ordered Sets for Sequence Functions E-4

Table E.2 Fibre Channel FC-0 Valid Combinations E-7

Table E.3 Nomenclature of FC-0 (cable plant speed-media-transmitter-distance) E-7

Table E.4 Fibre Channel Ordered Sets for Sequence Functions E-8

LIST OF FIGURES

Figure 1.1 TAXI-275 Transmitter Block Diagram 1-2
Figure 1.2 TAXI-275 Receiver Block Diagram 1-3
Figure 1.3 Basic TAXI-275 Operation 1-4
Figure 2.1 Valid Realignment Sequences 2-3
Figure 3.1 Power-Up Time to Lock 3-3
Figure 3.2 A Method of Synchronous Strobing 3-4
Figure 3.3 Transmitter Byte Latency 3-5
Figure 3.4 Receiver Byte Latency 3-5
Figure 4.1 Standard Load Circuit 4-2
Figure 4.2 Pull-Down with I_{OH} Matched to Standard Load 4-2
Figure 4.3 DC Coupled Termination 4-3
Figure 4.4 Pull-Down and Termination for AC Capacitor Coupled Link 4-4
Figure 4.5 TAXI Serial Link with Output Stage Turned Off by Transmission Line 4-4
Figure 4.6 Simple TAXI-275 Based Fiber Optic Link 4-5
Figure 4.7 DC Coupled TAXI Fiber Optic Interface (Unterminated) 4-6
Figure 4.8 DC Coupled TAXI Fiber Optic Interface (Terminated) 4-6
Figure 4.9 AC Capacitor Coupled TAXIchip to Optical Transceiver Interface 4-7
Figure 4.10 Coaxial Cable Interface 4-8
Figure 4.11 Shielded-Twisted Pair Cable Interface 4-9
Figure 4.12 Transformer Coupling of a 75 Ω Coaxial Cable Media 4-10
Figure 4.13 Transformer Coupling of Quality 150 Ω STP Media 4-10
Figure 4.14 Compensation Circuit for Copper Media 4-11
Figure 4.15 Receiver Side Logic for Detection of Cable Disconnect 4-13
Figure 5.1 Transmitter and Receiver Decoupling Schematics 5-2
Figure 5.2 Jogs and Glitches in the Clock Line 5-2
Figure 5.3 Decoupling Terminations 5-3
Figure 5.4 Fiber Optic Data Link Decoupling 5-4
Figure 6.1 Logic Diagram of 32-Bit Multiplexed Transmitter 6-3
Figure 6.2 Strobe Initiated Timing 6-4
Figure 6.3 ACK0 and ACK1 Timing 6-4
Figure 6.4 Logic Diagram of 32-Bit Demultiplexed Receiver 6-7
Figure 6.5 Timing Diagram of 32-Bit Demultiplexed Receiver 6-8
Figure B.1 A Typical ESCON Node Using TAXI-275 . . . . . B-2
Figure B.2 A Typical Fibre Channel Node Using TAXI-275 . . . . . B-2
Figure B.3 Block Diagram of a TAXI-275 ATM Node B-3
Figure E.1 ESCON Channel Switched Point-to-Point Topology E-1
Figure E.2 ESCON Channel Dynamic Connectivity without Single-Point-of-Failure E-2
Figure E.3 ESCON Channel Link Level Frame Format and Header E-3
Figure E.4 ESCON Channel Device Frame Format and Header E-5
Figure E.5 Fibre Channel Terminology and Switched Point-to-Point Topology E-5
Figure E.6 Fibre Channel FC-0 Open Fiber Control E-7
Figure E.7 Fibre Channel Primitive Sequence Protocol Example (Link Initialization) E-8
Figure E.8 Fibre Channel FC-2 Frame Format E-9
Figure E.9 Fibre Channel FC-2 Frame Header E-9
Figure E.10 Fibre Channel Physical Link Layers Class 1 and Class 2 . . . . . E-10
Figure E.11 Fibre Channel Building Block Hierarchy E-11

CHAPTER

1 TAXI-275 INTRODUCTION

Modern electronic systems move data from point-to-point across physical layer boundaries using either serial or parallel data links. Parallel data links provide fast data transfers and are compatible with most computer architectures. However, conventional parallel data links are burdened with cost/performance issues such as costly multi-conductor cables, crosstalk, Radio Frequency Interference (RFI), bit-to-bit skew and other concerns associated with multiple wire interfaces. Serial data links, although simpler and less costly, have not traditionally provided sufficient bandwidth to compete with the high data transfer rates of parallel links.

Recent technological advances have altered the cost/performance trade-off between serial and parallel data transfer techniques. A new chip set from Advanced Micro Devices offers a high performance integrated alternative to traditional serial/parallel data transfer techniques. The TAXI-275 chipset (Transparent Asynchronous Transmitter-Receiver Interface) provides the means to establish a transparent high speed serial link between two high performance parallel busses. The TAXI-275 chipset consists of a Transmitter, which takes parallel data and transmits it serially at up to 275 MBaud, and a Receiver, which converts the serial data stream back to parallel form. The TAXI-275 devices provide a simple parallel interface through a high speed serial link, while maintaining the data bandwidth required by the system.

The TAXI-275 chipset provides parallel, to serial, to parallel data conversion like other TAXI products, but with an 8B10B coding scheme, and at serial transmission rates between 175 MBaud to 275 MBaud. The coding scheme and transmission rates meet the needs of many applications, while making the TAXI-275 compatible with the physical layer requirements of IBM™ ESCON™, ATM, and ANSI Fibre Channel.

Information on designing with the TAXI-275 as well as a brief description of some of the applicable standards are contained in this document.

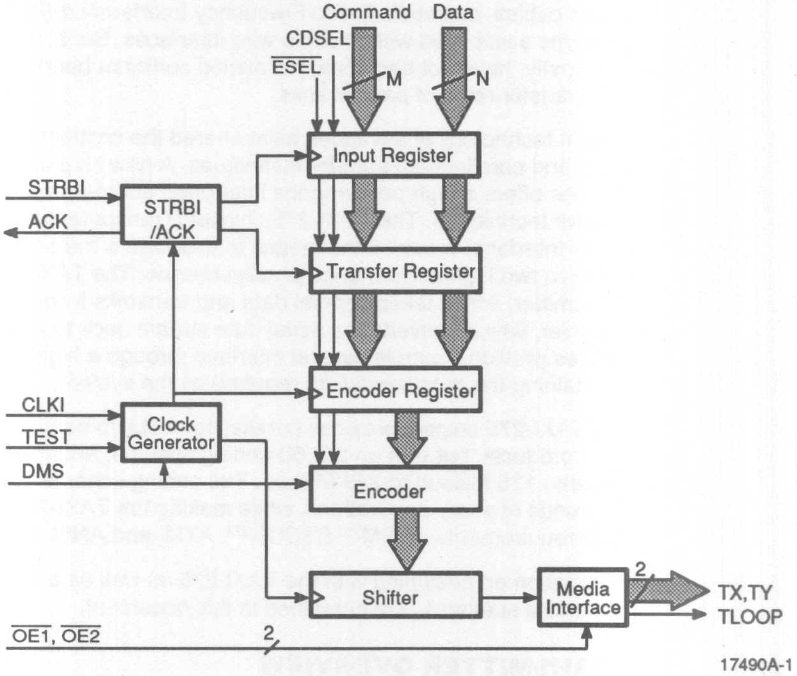
1.1 TRANSMITTER OVERVIEW

The TAXI-275 Transmitter consists of an input register, an encoder, a parallel to serial shift register, a multiplying Phase Locked Loop (PLL), and some control logic—see Figure 1.1 TAXI-275 Transmitter Block Diagram. Data is input to the register, encoded, and shifted out at the serial baud rate. The coding scheme used is the balanced 8B10B coding scheme specified for the ANSI Fibre Channel specification as well as for the ESCON and ATM descriptions. The encoding scheme divides an 8-bit byte into two nibbles, one 5-bit nibble and one 3-bit nibble. The 5-bit nibble is encoded into a 6-bit symbol and the 3-bit nibble is encoded into a 4-bit symbol. The new 10-bit encoded byte is formatted into an NRZ data stream for output to the media. The 8B10B encoding is 80% efficient with a 175 MBaud to 275 MBaud transmission rate to generate a data rate of 140 Mbps to 220 Mbps.

The Am79168 Transmitter has differential pseudo-ECL (Emitter Coupled Logic referenced to +5 V) outputs which are designed for 50 Ω impedance transmission lines. This capability makes it easy to directly interface to shielded twisted pair, coaxial cable or fiber optic modules. The Transmitter serial output lines are not limited to 50 Ω impedance cables, they can be interfaced to a variety of different impedance valued cables using the termination schemes recommended in Section 4, Interface to the Serial Media of this manual.

The pseudo-ECL interface using fiber optic modules has a few advantages over using copper media. In addition to providing high bandwidth and low attenuation, fiber optic data transmission also offers noise immunity, eliminates RFI and provides data security. Declining optical components costs are bringing the advantages of fiber optic data transmission to an ever wider range of applications from process control to avionics. The TAXI-275 chipset is the ideal complement for fiber optic interfaces.

Figure 1.1 TAXI-275 Transmitter Block Diagram



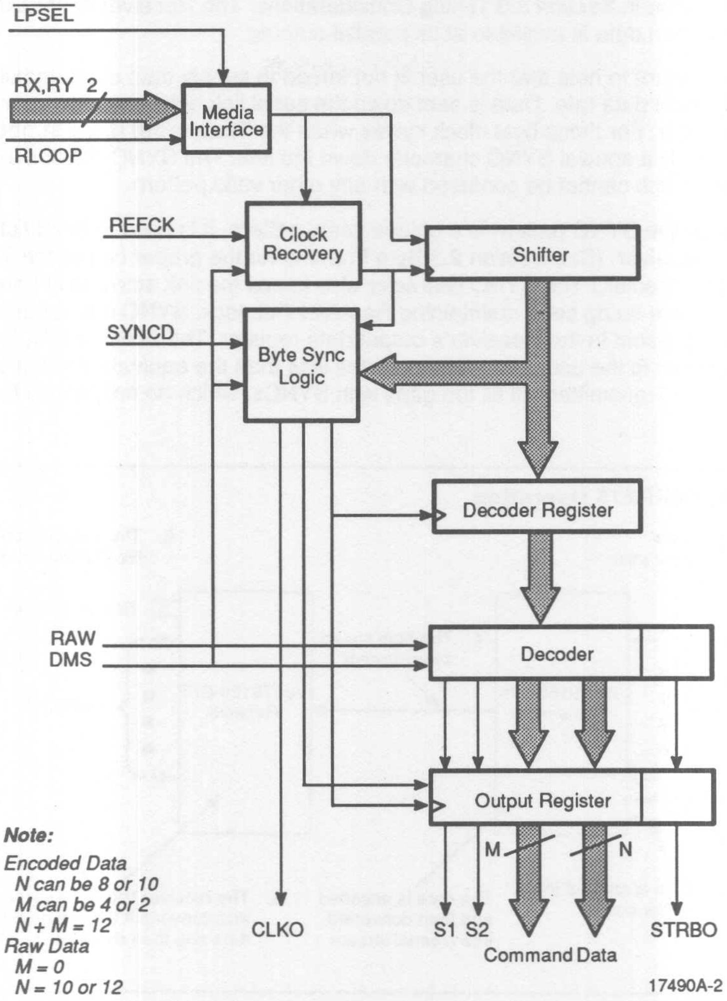
Note:

Encoded Data
N can be 8 or 10
M can be 4 or 2
N + M = 12
Raw Data
M = 0
N = 10 or 12

1.2 RECEIVER OVERVIEW

The TAXI-275 Receiver accepts the 8B10B encoded data stream into a serial-to-parallel converter, decodes and then outputs the received data to a parallel bus along with accompanying command signals—see Figure 1.2 TAXI-275 Receiver Block Diagram. Clock recovery of the input serial data stream is performed by an on-chip data tracking PLL.

Figure 1.2 TAXI-275 Receiver Block Diagram



Note:
Encoded Data
N can be 8 or 10
M can be 4 or 2
N + M = 12
Raw Data
M = 0
N = 10 or 12

1.3 USING THE TAXI-275 CHIPSET

The TAXI-275 chipset has a serial baud rate range of 175 Mbaud to 275 Mbaud. In 8-bit mode, this corresponds to a parallel data transfer rate of 17.5 Mbyte/s to 27.5 Mbyte/s. In 10-bit mode, the corresponding parallel data rate is 14.6 Mbyte/s to 22.9 Mbyte/s.

Encoding data with the TAXI-275 resident 8B10B transmission code provides 80% data throughput efficiency. Maximum data throughput in 8-bit encoded mode is 220 Mbit/s or 80% of the maximum serial transmission rate of 275 Mbaud.

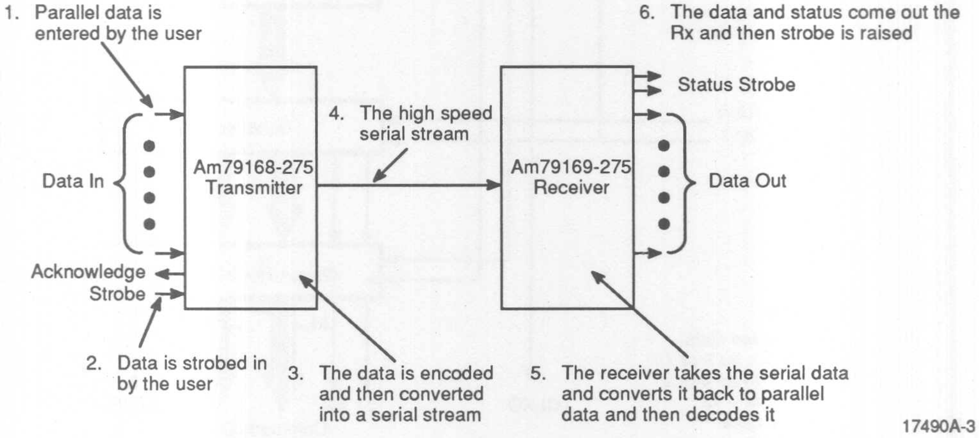
Because the high speed encoding/decoding and serial-to-parallel conversions are transparent to the user, the chipset can be visualized as a data register that accepts parallel data on the Transmitter side, and outputs the same parallel data on the Receiver side. A system designer would see an effective parallel transfer rate of up to 27.5 Mbyte/s.

Parallel input data is accepted into the TAXI-275 Transmitter with a simple Strobe/Acknowledge handshake. Asynchronous and synchronous data strobing are both possible and covered in Section 3.0 Timing Considerations. The Receiver asserts an output strobe when data is available at its parallel outputs.

It is important to note that the user is not forced to supply data at the maximum byte rate of the serial data rate. Data is sent down the serial link only when the user strobes the Transmitter. For those byte clock cycles when the Transmitter is not strobed, it automatically sends a special SYNC character down the link. The SYNC character is a unique bit pattern which cannot be confused with any other valid pattern.

Because the SYNC pattern is a unique serial pattern, it is used to establish byte framing at the Receiver. (See Section 2.3 Byte Framing for the proper use of the TAXI-275 SYNC character.) The SYNC character also keeps the link active when no other symbols are being sent, maintaining Receiver PLL lock. SYNC will not over-write data already present in the Receiver's output data register. The serial rate is, therefore, truly transparent to the user; at input data rates less than the equivalent serial bit rate, the TAXI-275 Transmitter will fill the gaps with SYNCs, which do not disturb Receiver output data.

Figure 1.3 Basic TAXI-275 Operation


1. Parallel data is entered by the user
6. The data and status come out the Rx and then strobe is raised
Status Strobe
4. The high speed serial stream
Data In
Am79168-275 Transmitter
Am79169-275 Receiver
Data Out
Acknowledge
Strobe
2. Data is strobed in by the user
3. The data is encoded and then converted into a serial stream
5. The receiver takes the serial data and converts it back to parallel data and then decodes it
17490A-3

1.4 DATA WIDTHS, DATA AND COMMANDS

The Am79168-275 TAXI Transmitter and Am79169-275 TAXI Receiver interface directly to an 8-bit or 10-bit data bus. Each TAXI-275 has 12 parallel interface lines which are designated as either command or data bits. Command bits implement user defined system supervisory functions, such as Initialize Your System or Re-try which cannot be embedded in the data path.

Two different widths are possible: 8 data bits and 4 command bits or 10 data bits and 2 command bits. This choice of data and control bus widths allow flexibility to meet different system bus width requirements, while providing the capability of merging control and data into a common data stream.

1.5 OPERATIONAL MODES—NORMAL, RAW, LOOPBACK, SYNCD, TEST

A TAXI-275 based link can be operated in a number of modes. These modes are Normal operation, Raw data input operation, Loopback testing, and Synchronization disable operation. Normal operation is used in systems that use 8-bit or 10-bit data that is required to be encoded, transmitted serially in Non Return to Zero (NRZ) format to a Receiver that will decode the data and output the same data that was transmitted. Raw data operation is used with 10-bit or 12-bit data that does not need to be encoded on the Transmit end or decoded on the Receive end. In Raw data operation an alternate form of coding should be used that does not violate the 8B10B transition density requirements of the Receiver PLL. Loopback operation is used to evaluate a single node to ensure that it is transmitting and receiving data properly. SYNCD operation is used when resynchronization of the Receiver to the K28.5 character is not desired. Test mode is for factory testing and is not recommended for use in an active link.

CHAPTER

2 DATA ENCODING OVERVIEW

Most serial data transmission requires some form of encoding before the data is output to the transmission medium. Encoding is the process of converting a set of data bits m to a set of code bits n.

The main purpose of the encoding operation is to include clock information in the data stream. Without this timing information, the Receiver would not be able to distinguish adjacent bits of the same value. For example, if we transmit 1000 logic ones followed by one or two zeros. An accurate clock is needed to tell the Receiver when to sample the incoming bit stream to determine if the bit is a one or a zero. Since the Transmitter and Receiver have only one data path between them, the clock timing information must be included in the serial data stream.

The TAXI-275 chipset uses 8B10B or 10B12B coding, so that m is 8 or 10 and n is10 or 12. In 8-bit mode, one 5-bit nibble is converted to a 6-bit code and one 3-bit nibble is converted to a 4-bit code to form one 10-bit encoded character. In 10-bit mode both encoders are replaced with 5B6B encoders, yielding a 12-bit encoded character.

To aid in error detection, the TAXI-275 coding scheme utilizes Disparity and Running Disparity (RD). Each symbol of a encoded character has an associated Disparity. The Disparity has to do with the sum of ones and zeros in the symbol. The Running Disparity or sum of previous Disparities is used to determine the next symbol to be transmitted or Received. If the Receiver does not receive a character with the expected disparity, an error is flagged by the violation logic. Refer to Appendix D, Data Encoding, for more detailed information.

The TAXI-275 can encode two types of information, Data or Commands. Commands are special symbols which are typically used as control functions at the Receiving end of the link. Commands may be four or two bits wide corresponding to the TAXI-275 device data widths of eight or ten bits, respectively. The state of the CDSEL input at the time of strobe determines whether a Command or Data is to be transmitted. The Command bits are encoded into 10-bit or 12-bit symbols which are special cases of the 8B10B or 10B12B code which are not used for data.

In the absence of a strobe pulse a unique symbol (SYNC) is automatically generated to maintain link synchronization.

The TAXI-275 chipset coding scheme generates NRZ (Non Return to Zero) serial data. A logic one is represented as a high level and a logic ZERO is represented as a low level. This form of serial data in combination with the encoded symbol patterns ensure that the Receiver PLL will get the required transition density to maintain frequency lock. The number of transitions is important since the Receiver PLL can only make a phase comparison and initiate a correction at a transition. The required minimum number of transitions helps keep the PLL solidly in lock.

A detailed description of the coding scheme along with the complete 8B10B coding table is shown in Appendix D, Data Encoding.

2.1 UNENCODED (RAW) DATA TRANSMISSION

Unencoded or raw data transmission bypasses the TAXI-275 device resident 8B10B and 10B12B encoding schemes and allows for data words either 10 bits or 12 bits wide. Raw transmission is useful for either forcing serial violations during system diagnostics or for utilizing encoding schemes other than 8B10B or 10B12B. Some encoding is usually required to maintain Receiver synchronization. If alternative encoding schemes are used, the designer must ensure that the transition density is sufficient to keep the Receiver from losing byte alignment; the transition density rules for 8B10B encoding should be followed.

The Transmitter and Receiver both have raw data operation capabilities. The Transmitter uses the $\overline{\text{ESEL}}$ input to allow transmission of raw data. The $\overline{\text{ESEL}}$ input is captured by the rising edge of STRBI. Therefore, the $\overline{\text{ESEL}}$ input can be used to select on a byte-by-byte basis whether to send encoded or unencoded data. The RD is suspended on the Transmitter and does not change value while unencoded data is being transmitted.

Hardwired raw data operation is also possible on the Transmitter by hardwiring the $\overline{\text{ESEL}}$ input High and connecting the CLKI signal to the STRBI input. Connecting the CLKI to STRBI input ensures that a no-strobe condition does not take place. If the Transmitter experiences a no-strobe condition, a resynchronization character (K28.5 in 8-bit mode) will be transmitted even if the $\overline{\text{ESEL}}$ input is hardwired to disable encoding. In hardwired raw data operation, a user determined default data pattern must be used on the ten Data inputs. The default pattern will have to be transmitted when data packets are not intentionally being transmitted by the system. This ensures that the transition density requirement of the PLL is not violated.

The Receiver has the capability to operate in two hardwired modes, Raw Mode and Encoded Mode. The RAW input on the Receiver is not designed to be changed while the Receiver is operating. The raw data and encoded data modes are hardwired options only. As a special note, even though the Receiver is in Raw Mode, the Receiver will still byte align to a resynchronization symbol pair (K28.5s in 8-bit mode). If realignment is not desired, the Synchronization Disable (SYNCD) input can be used to disable the realignment capability.

To test the robustness of a system, the Transmitter can be used to transmit violation symbols on a byte-by-byte basis. The use of the $\overline{\text{ESEL}}$ input or raw data operation allows the user to intentionally transmit any violation symbol or sequence of violation symbols desired. The Receiver operates in the normal Encoded Mode during the testing process. Using raw data operation a variety of intentional errors can be forced through the Transmitter to the Receiver from one invalid symbol to a miscellaneous realignment sequence. These errors will be flagged by the violation logic.

The Receiver calculates and checks the RD value for each of the 3B4B and 5B6B nibbles of the byte being received. Therefore if the coding violations transmitted in raw data operation of the Transmitter changes the RD at the nibble level or byte level to an invalid value, the following RD values could be different from the Transmitter values and could cause violations to continue to occur even after the Transmitter is transmitting data in encoded mode.

2.2 VIOLATION DETECTION

A Violation is indicated when an incorrect code or Running Disparity is received by the TAXI-275 Receiver. The S1 and S2 status outputs of the TAXI-275 Receiver indicate the violation by bringing S1 Low and S2 High. The inherent error detection of the TAXI-275 is based on the 8B10B encoding scheme.

The violation detection feature does not indicate all errors. The only errors that are indicated are incorrect data symbols, command symbols and RD errors. It is possible that a data symbol could be corrupted into a different but still valid data symbol as explained in the 8B10B Coding Section. The RD error detection adds an extra level of error indication that enables the detection of some of the transformed data symbols. However, it still does not detect all of these errors. An additional form of error detection must be implemented in fault sensitive systems.

2.3 RECEIVER BYTE FRAMING (USE OF SYNC CHARACTER)

To receive encoded data correctly, the TAXI-275 Receiver must properly align with the data stream byte boundaries. The TAXI-275 device uses the K28.5 character, which has five consecutive ones following two zeros, as the realignment characters. Two consecutive K28.5 characters or two K28.5 characters separated by up to three 10-bit words are required to align the proper byte boundaries (See Figure 2.1 Valid Realignment Sequences). The possibility of an alias SYNC character is reduced by several orders of magnitude by requiring two realignment characters.

Figure 2.1 Valid Realignment Sequences

. . . | XX | K28.5 | K28.5 | XX | . . .

. . . | XX | K28.5 | XX | K28.5 | XX | . . .

. . . | XX | K28.5 | XX | XX | K28.5 | XX | . . .

. . . | XX | K28.5 | XX | XX | XX | K28.5 | XX | . . .

Note:
XX denotes any 10-bit pattern

17490A-4

If even further guarantees against alias SYNCs are required, the SYNCD input of the Receiver can be used to disable the byte realignment function of the Receiver. The SYNCD input when driven High prevents the Receiver from checking for a realignment sequence. The SYNCD input is a design feature that is provided for use mainly in Raw Mode operation. When a design requires an encoding scheme other than 8B10B that does not use the K28.5 symbol as the realignment character, the realignment function can be suppressed. In this situation the realignment function will have to be designed external to the Receiver device.

Another use of the SYNCD input is to ensure that a data packet corruption will not cause a miscellaneous byte realignment. The possibility of a miscellaneous realignment is very small with the TAXI as described above, but some designers may still desire an even smaller possibility of error. A way to achieve this is to raise the SYNCD input after the successful start of transmission of a data packet, disabling the realignment capabilities of the Receiver. The data packet is then safe from any possibility of a byte boundary realignment. At the end of the data packet being transmitted, the SYNCD input will be lowered re-enabling the byte realignment capabilities of the Receiver. In this design, the only loss of data will be the actual bytes that were corrupted, thus protecting the remainder of the data packet being transmitted.

...tion detection in that there can be a number of erro...
...ed are incorrect, data and control (sync) symbol...
...data symbol could be corrupted into a different but a...
...ed in the 5B/6B Coding Scheme. The RD error d...
...detection that enables the determining some of the...
...ver, it will does not detect all of these errors. An add...
...be implemented in local memory systems.

...CEIVER BYTE FRAMING (USE OF SYN...

...ive encoded data correctly, the TAXI-275 Receive...
...correct byte boundaries. The TAXI-275 device uti...
...characters are one following two bands; at the recep...
...8.5 is considered as an K28.5 character requiring...
...ed in order the proper byte boundaries TAXI-275...
...ence). The possibility of an error SYNC character...
...ous as required best alignment character).

...Alignment Sequences

| XX | K28.5 | K28.5 | XX |

| XX | K28.5 | XX | K28.5 | XX |

| YY | K28.5 | XX | YY | K28.5 | YY |

| XX | K28.5 | XX | XX | YY | XX | K28.5 | XX |

...n if other placements control data SYNCs are at...
...ver can be used to handle the byte realign and...
...D input when it have high prevents the Receiv...
...ence. The SYNC input is a design feature that...
...e operation. When a design requires an error fr...
...choose the K28.5 symbol as the realignment co...
...e guaranteed. In this situation the realignment fu...
...nal to the Receiver device.

...ver use of the SYNC input to prevent errors...
...g extraneous byte realignment. The possibility of...
...d with the TAXI as described above, the same as...
...ular possibility of error. A way to solve this is to...
...cessful start of transmission of a data packet into...
...ng Receiver. The data packet is frequently, and...
...alignment. At the end of the data packet being transm...
...ence re-enabling the byte realignment function...
...ytes of data will be the actual symbols that were...
...mainder of the data packet being transmitted.

CHAPTER

3 TIMING CONSIDERATIONS

3.1 PHASED-LOCKED LOOP

3.1.1 Introduction

The TAXI-275 PLL has been optimized to allow correct data recovery in the presence of the largest jitter possible. To this end, the PLL parameters, most notably loop bandwidth, have been chosen to enhance the jitter tolerance of the Receiver.

This jitter optimization is achieved at the expense of lock-up time. Fortunately in TAXI-275 systems, lock-up time is relatively unimportant, since the system must only achieve lock during system power-up. The lock-up time will be related to the start-up characteristics of the system power supply. (See Section 3.1.2 and 3.1.3 Data Acquisition Time.)

The actual time to lock begins during power-up, when both the Transmitter and Receiver and the entire link are approaching full V_{CC} (See Section 3.1.3 Data Acquisition for details). Transient effects, other than PLL characteristics, which typically occur during power-up can either lengthen or shorten the apparent lock time. These effects are a function of actual implementation and are not discussed here. The discussion which follows assumes that both Transmitter and Receiver are fully powered, and that the link is fully operational. The only effects included are PLL transient effects.

If there is no data on the link (if the Transmitter is off, or if there is a quiet line) the data recovery PLL will drift to its natural oscillation frequency. This frequency is determined by component values and tolerances inside the PLL, and will vary slightly from both the reference frequency (REFCK of the Receiver) and the Transmitter data frequency (CLKI of the Transmitter).

When data is first received on the serial lines, the receive PLL must achieve phase lock from its resting frequency. The structure of the PLL used in the TAXI-275 device ensures that this resting frequency will be no more than 0.7% from the reference frequency applied at REFCK. This is in addition to the specified Transmitter/Receiver frequency mismatch allowed by the crystal tolerance specification of ±0.1%.

Neglecting frequency variations in the Transmitter and jitter in the data stream, the time to lock is related to the PLL loop bandwidth and damping factor and to the transition density. The loop parameters are set by the internal component values and tolerance of the TAXI-275 chipset. The 8B10B coding scheme balances the number of ones and zeros so that one particular pattern does not significantly effect the lock time of the Receiver unlike the previous TAXI products using the 4B5B coding scheme.

3.1.2 Data Acquisition Time (Quiet to Lock Time)

Probably the best method to determine the time for the TAXI-275 to acquire PLL lock is to check for proper byte alignment. The time it takes for the Receiver to acquire PLL lock and byte alignment from a quiet condition at the Receiver inputs is simple to verify.

The Transmitter can supply a quiet line state and an active line state to the Receiver serial inputs. The $\overline{OE1}$ input on the Transmitter when enabled allows data to be

The Transmitter can supply a quiet line state and an active line state to the Receiver serial inputs. The $\overline{\text{OE1}}$ input on the Transmitter when enabled allows data to be transmitted on the serial outputs. When the $\overline{\text{OE1}}$ input is disabled, a quiet line state exists on the serial lines.

The Receiver indicates byte alignment with the S1 and S2 status outputs. When the Receiver is not locked or is in the process of obtaining lock, the status outputs will indicate violations, S1 = Low and S2 = High. The status output when byte alignment is achieved will be the realignment, S1 and S2 High status. Immediately following this status will be valid Command symbols, S1 = High and S2 = Low. The Transmitter should only transmit the default K28.5 character in this test case. Therefore, PLL lock/byte alignment are indicated by the stable S1 = High S2 = Low states on the status outputs.

When the PLL attains frequency lock, the TAXI-275 device will byte align within about two byte times when using the K28.5 symbol. The time from the Transmitter $\overline{\text{OE1}}$ input going from High to Low until the Receiver status outputs indicate a stable S1 = High, S2 = Low is the time to data acquisition or time to lock of the Receiver. The time to lock using the above test method has been found to be typically less than 1 ms.

3.1.3 Data Acquisition Time (Power-Up to Lock Time)

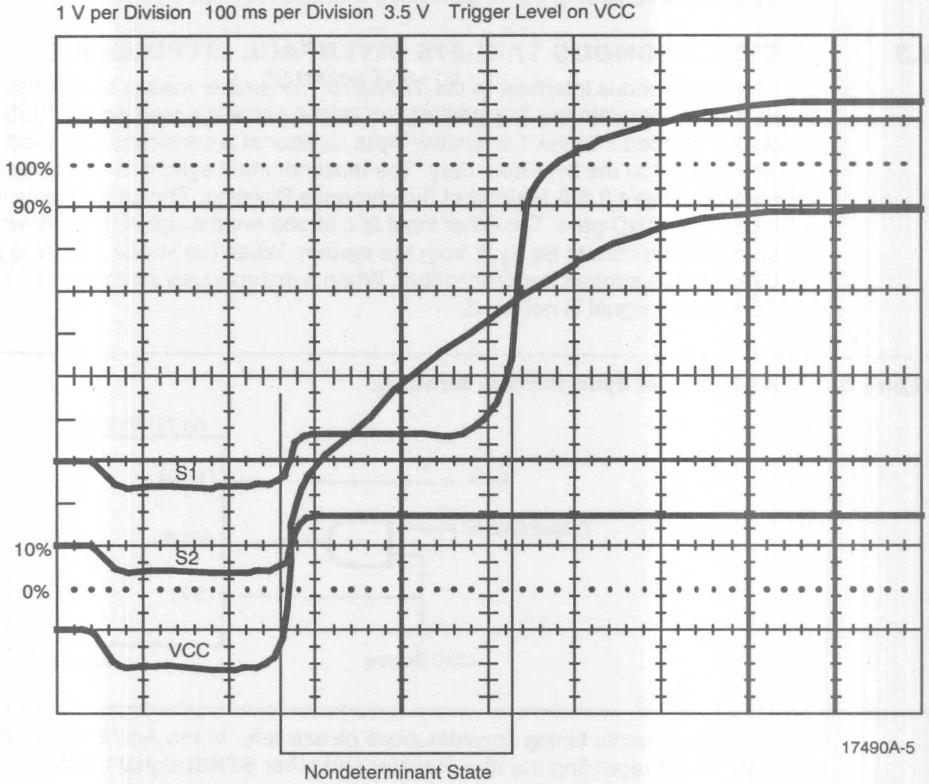
The time it takes for the Receiver to acquire PLL lock and Byte alignment from the time of power-on may be of some concern to some system designs. The best way to measure the time to lock in this case is to capture the power-up sequence while observing the S1 and S2 outputs of the Receiver. As in the case above the S1 = High and S2 = Low outputs will be stable when the Transmitter and Receiver are operating properly.

The setup used for the power-up test was as described below. An evaluation board was configured to run in Loopback mode with the Transmitter hardwired to send the default nostrobe K28.5 pattern in 8-bit mode. The setup voltage was set for 5.0 volts. The probe for the V_{CC} supply was connected at the evaluation board power connector. The probes for the S1 and S2 signals were connected to the respective evaluation board connector pins.

When operating properly, the Receiver would detect the K28.5 character as a command and output the command status on the S1 and S2 output pins as indicated above, thus signifying that PLL lock and byte alignment had occurred. The TAXI-275 devices used in the evaluation were nominal parts selected from laboratory stock.

As expected the TAXI-275 devices achieved PLL lock and byte alignment before the power supply had completed power-up. As soon as the power to the test system reached approximately 3.5 volts the TAXI-275 devices were operating properly as illustrated in Figure 3.1 Power-Up Time to Lock.

Figure 3.1 Power-Up Time to Lock



3.2 ASYNCHRONOUS TAXI-275 INTERFACES (STRBI/ACK)

One of the asynchronous interface methods of the TAXI Transmitter is the STRBI/ACK method. The STRBI/ACK method is used for designs where the strobe pulse that captures data for the Transmitter does not have a consistent relationship with respect to the Transmitter clock (CLKI input); data is strobed into the Transmitter input register at random locations with respect to the byte-boundary. When a strobe pulse occurs, the next byte of data to be transmitted must not be strobed into the Transmitter until the current byte of data has cleared the input register. The ACK signal indicates to the external system that the current byte has cleared the input register and the next byte of data can be strobed into the Transmitter. When data is not strobed into the Transmitter during a byte-time (no strobe pulse between byte boundaries) a default K28.5 character is transmitted.

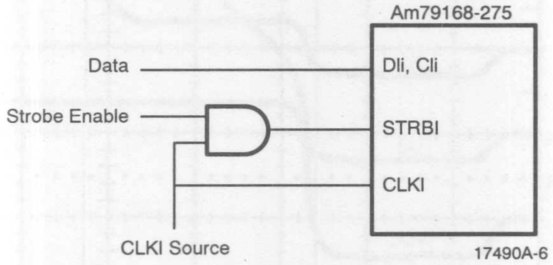
Another asynchronous interface method involves only the STRBI input. When the frequency of data strobed into the Transmitter is less than the clock frequency of the Transmitter CLKI input, then data can be continuously strobed to the Transmitter without the use of the ACK signal indication. To prevent the loss of data, it must be ensured that the data strobe rate does not equal or exceed the Transmitter clock source rate. Because the strobe rate or frequency is slightly slower than the Transmitter operating frequency occasionally a byte boundary will pass without a strobe pulse occurring causing a K28.5 to be transmitted. It must be noted that at power up and at any time the byte alignment on a link is lost, a minimum of a pair of K28.5 (CMD5) characters must be transmitted to realign the byte boundary for the Receiver.

For more specific timing considerations please refer to the Am79168/Am79169-275 Data Sheet regarding the t6 parameter and other STRBI signal timing.

3.3 SYNCHRONOUS TAXI-275 INTERFACE (STRBI/CLKI)

The synchronous interface to the TAXI-275 Transmitter means by definition that when data is strobed into the Transmitter, the strobe pulse is based on the CLKI input source; data is strobed into the Transmitter input register at a consistent and predictable location with respect to the byte-boundary. The most common synchronous design is shown below in Figure 3.2 A Method of Synchronous Strobing. The CLKI source is used as one input to an AND gate. The other input is a strobe enable signal that follows the same timing as the data to be input from the system. When the strobe enable is High, the CKLI source strobes the Transmitter. When synchronously strobing the Transmitter the ACK output signal is not used.

Figure 3.2 A Method of Synchronous Strobing



For more specific timing considerations please refer to the Am79168/Am79169-275 Data Sheet regarding the t6 parameter and other STRBI signal timing.

3.4 TX/RX LATENCY

The latency of the Transmitter is the time from the rising edge of the STRBI signal associated with a byte to the time the first bit of that byte is output on the serial lines. The latency through the TAXI-275 Transmitter is typically the time, T1 from the STRBI rising edge to the CLKI falling edge plus 8-bit times plus a fixed delay of approximately 3 ns as illustrated in Figure 3.3 Transmitter Byte Latency. The fixed delay is due to internal buffering and does not change over frequency.

The only situation where the latency could be greater is if the Transmitter is strobed in the t6 zone. When a byte is strobed into the Transmitter in the t6 zone it is uncertain whether the byte will start transmission in the current byte time or the following byte time. If the byte starts transmission in the following byte time, the time T1 will be increased by one byte time as illustrated in Figure 3.3 Transmitter Byte Latency.

The latency of the Receiver is the time from when the first bit of a byte reaches the serial inputs to the time when that complete byte is output on the Data/Command lines. The latency through the TAXI-275 Receiver is typically two and a half byte time periods as illustrated in Figure 3.4 Receiver Byte Latency.

Figure 3.3 **Transmitter Byte Latency**

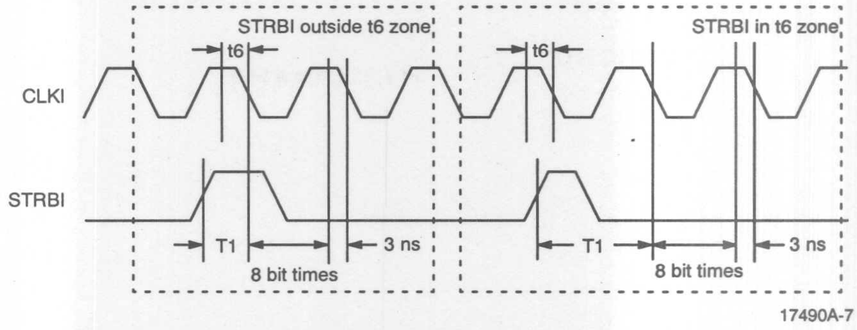
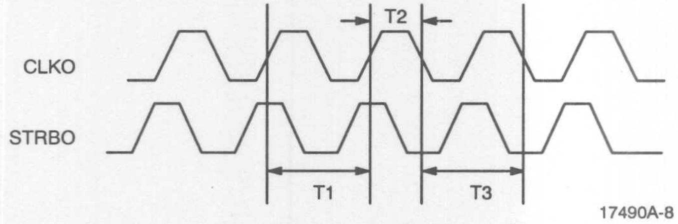


Figure 3.4 **Receiver Byte Latency**



Notes:

T1 Encoded 10 bits shifted into Receiver

T2 Encoded data captured for Decoder Logic

T3 Data decoded and clocked into output register

CHAPTER

4 INTERFACE TO SERIAL MEDIA

4.1 COAXIAL AND FIBER OPTIC MODULE INTERFACE

4.1.1 Introduction

The Am79168/Am79169-275 TAXI-275 chipset is capable of providing a high speed point-to-point serial link over fiber-optic, coaxial, or twisted pair media. The choice of an appropriate media depends primarily on line length and data rate. This chapter discusses the issues involved in media choice and TAXI-275 device requirements for driving different types of media.

Any TAXI-275 media interface design must first take into account the electrical properties of the TAXI-275 Transmitter and TAXI-275 Receiver. The Transmitter serial output drivers are open emitter, emitter followers which generate Pseudo-ECL (PECL) levels when terminated by pull-down resistors to a voltage more negative than V_{OL}. PECL is ECL referenced to the +5 V supply, so that typically V_{OH} = (5–0.8) and V_{OL} = (5–1.8) V. The Receiver input will switch on a 50 mV differential input voltage and has a large common mode range. The average DC value of the input signal is therefore relatively unimportant.

There are four broad classes of TAXI-to-media interface:

- Very short (<1.5" link length), DC coupled
- Terminated, DC coupled
- Terminated, AC capacitor coupled
- Terminated, AC transformer coupled

The very short link is typically between an optical component and a TAXI. The AC terminated cases are used for driving cables or optical components with incompatible power supply and/or logic levels. Transformer coupling is a form of AC coupling that is discussed in a separate section; it is the recommended termination scheme (refer to Section 4.3).

4.1.2 Very Short Links

For inter-connections in which the distance between the TAXI (Transmitter or Receiver) and the optical link is less than 1.5", transmission line terminations are not necessary. The effect of mis-match is distortion and slowing of the transition due to the reflections. All that is required is an appropriate PECL pull-down resistor R_E. Elimination of reflections is not required for short line lengths because the round-trip time of the media is significantly less than the 1.2 ns TAXI rise and fall time.

Figure 4.1 **Standard Load Circuit**

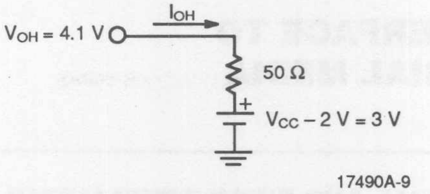
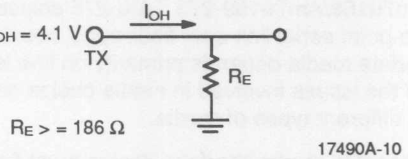


Figure 4.2 **Pull-Down with I_{OH} Matched to Standard Load**



The lower limit for R_E is that value which produces the maximum value of I_{OH}. In a standard PECL load circuit (Figure 4.1), I_{OH} max is given by:

$$(V_{OH}–(V_{CC}\text{-}2))/50 = (4.1–3)/50 = 22\text{ mA}$$

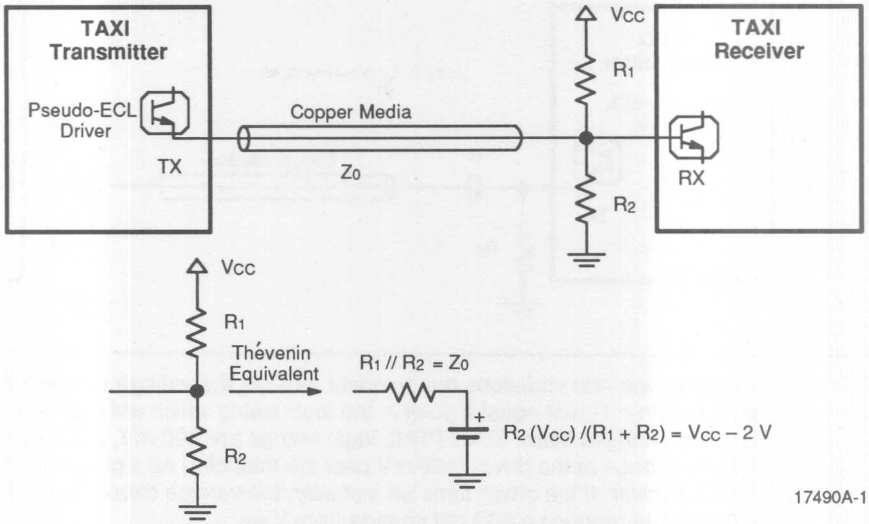
If we return R_E to ground instead of 3 V (Figure 4.2), the minimum value of R_E becomes 4.1V/22 mA, or 186 Ω.

Note that the supply voltage and logic level of the optical components must match those of the TAXI in order for the DC connection to work. If the supply voltage or the logic levels are incompatible, an AC connection must be used.

4.1.3 Terminated, DC Coupled

The termination shown in Figure 4.3 may be used for direct connection of a TAXI Transmitter to a TAXI Receiver. The termination provides both the line termination ($R1//R2 = Z_0$ = line characteristic impedance) and a pull-down voltage. The Thevenin equivalent of this termination looks like Z_0 pulled down to V_{CC} – 2 V, assuring both matched termination and adequate V_{OL}. Using V_{CC} and a voltage divider provides the pull-down voltage without the need for a separate power supply.

Figure 4.3 DC Coupled Termination

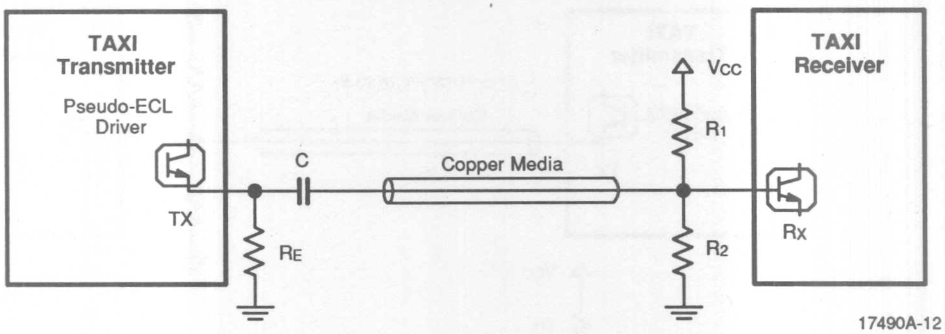


4.1.4 Terminated, AC Capacitor Coupled

The typical arrangement for a capacitor coupled link is shown in Figure 4.4. R_E is returned to ground to provide the PECL (pseudo-ECL) pull-down for the driver. The capacitor C blocks the DC, and R_1 and R_2 terminate the line and provide a DC bias level for the Receiver. Since only AC variations are passed through the coupling capacitor, the bias level at the termination should be set to the midpoint of the signal swing of the Receiver, to equalize the voltage drop in both High and Low states. **Note that this bias level is not the same as that which is recommended for the DC coupled case.**

The minimum value of R_E was previously established as 186 Ω, to avoid exceeding I_{OH} max. The maximum value of R_E must be small enough to supply the transmission line with enough current to avoid cutting off the output driver. When switching from the High to the Low state (see Figure 4.5), the transmission line may cause the emitter voltage of the driver to fall slower than the base voltage, causing the output transistor to turn off. When the output transistor turns off, its output impedance becomes very high, increasing the RC time constant, and slowing the falling edge rate. This variation in edge rate is significant only until the falling edge crosses the threshold level of the receiver's differential amplifier.

Figure 4.4 **Pull-Down and Termination for AC Capacitor Coupled Link**



To avoid edge rate variations due to driver turn-off, the voltage to which the driver is taken at turn-off must equal a point in the logic swing which will guarantee that the Receiver changes state. Since PECL logic swings are 800 mV, it is safe to choose a 500 mV change at the driver (100 mV past the midpoint) as a guaranteed state change at the Receiver. If the driver turns off instantly, the voltage divider formed by R_E and Z_0 is required to produce a 500 mV change from V_{OH}.

$$V_{OH} \times R_E/(R_E+Z_0) = V_{OH}-0.5$$

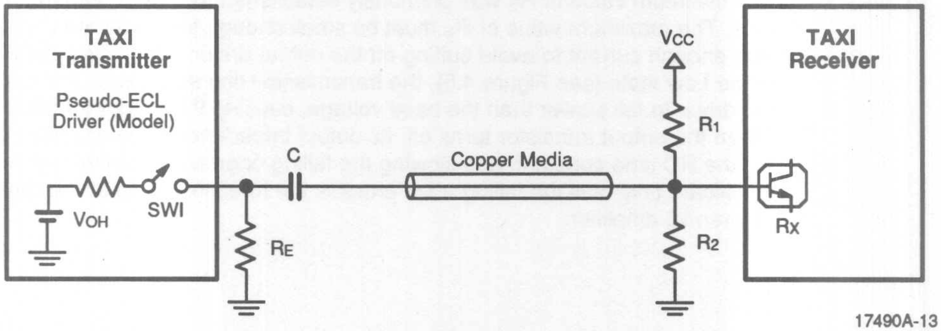
$$4.1 \times R_E/(R_E+Z_0) = 3.6$$

$$R_E = 7.33\ Z_0$$

As a general rule:

$$186 \leq R_E \leq 7.33\ Z_0$$

Figure 4.5 **TAXI Serial Link with Output Stage Turned Off by Transmission Line**



Notes:

Closed: Driving Transistor is ON

Open: Driving Transistor is OFF

$R_E \leq 7.33Z_0$

4.1.5 Baseline Wander and AC Capacitor Coupling

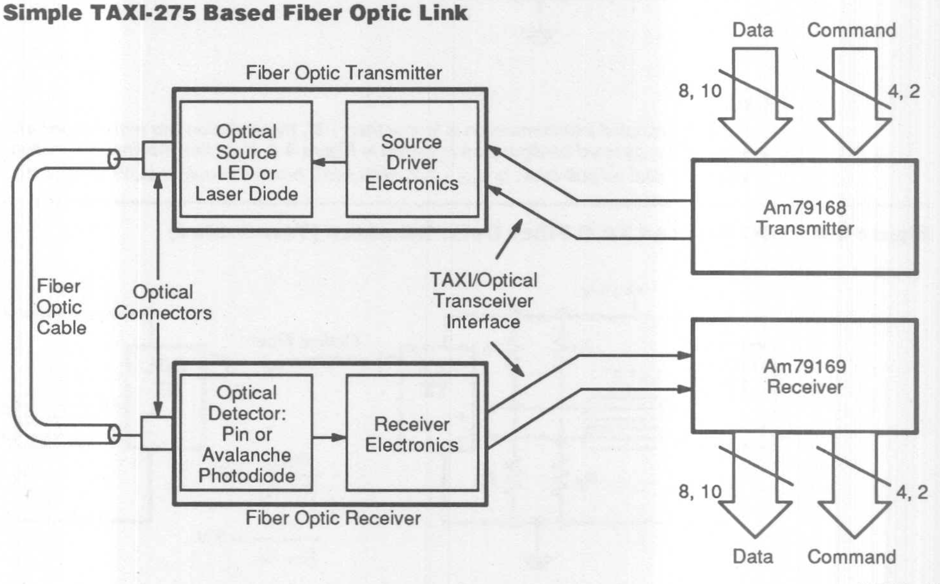
The nature of the 8B10B encoding scheme minimizes baseline wander (DC offset) for any data patterns. This is unlike the 4B5B encoding scheme used in previous TAXI products. The 8B10B encoding method allows any single data pattern or group of data patterns to be transmitted in AC coupled environments without difficulties of DC offset. The 8B10B coding scheme is DC balanced.

4.1.6 Interfacing to Fiber Optic Transmitters/Receivers

The TAXIchip set can be used in conjunction with optical components and optical fiber to form a simple fiber optic communications link. Optical transmission has many advantages over copper media transmission including immunity to EMI/RFI, low attenuation, electrical isolation, data security, and wide bandwidth. Because of these features, the use of optical fiber as the serial medium will result in improved performance of the TAXI link. Depending on the type of fiber and the optical components used, TAXI links using optical fiber can cover distances of several miles.

Figure 4.6 shows a block diagram of a complete TAXI fiber optic link. Optical components can be obtained from one of the sources listed in Appendix C.

Figure 4.6 Simple TAXI-275 Based Fiber Optic Link



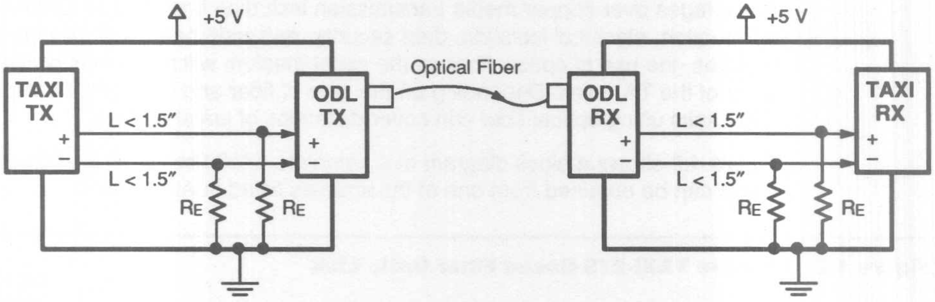
17490A-14

4.1.7 DC Coupled TAXI-275 Fiber Optic Interface

When passing data between the TAXI and an optical module, care must be taken to assure that the logic levels of the TAXI and the optical components are matched. If the supply voltages of the optical components do not match those of the TAXI, the logic levels will differ and the interface will require AC coupling to isolate them. If the optical components and TAXI-275 power requirements are the same, and if the two components are connected to the same power and ground planes, then a DC coupled interconnection may be desired.

For DC coupled interconnections in which the distance between the TAXI and the optical module is less than 1.5", transmission line terminations may not be necessary. All that is required is the appropriate ECL pull-down as shown in Figure 4.7. On these short line lengths, elimination of line reflections is not critical. However, without any increase in complexity or power consumption, line reflections can be reduced simply by locating the pull-down resistor, R_E, at the end of the line instead of at the beginning. This reduces the reflection coefficient at the end of the line, and therefore reduces the magnitude of the reflections.

Figure 4.7 DC Coupled TAXI-Fiber Optic Interface (Unterminated)

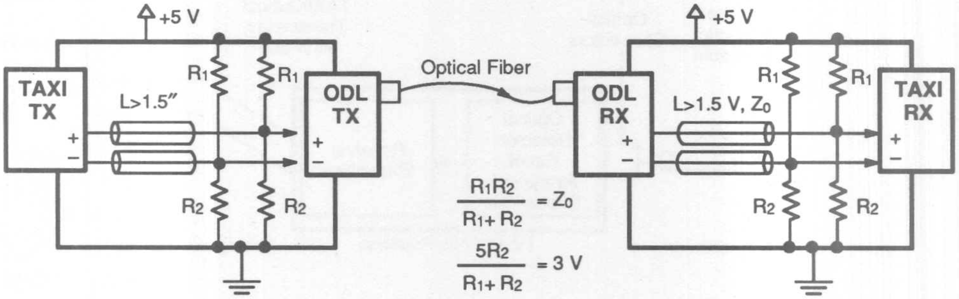


17490A-15

Note:

1. *If the DC-coupled interconnection is longer than 1.5″, transmission line terminations are required. For this case, the suggested configuration is shown in Figure 4-8. Note that the line termination network also provides the desired pull-down to V_{CC} – 2 V, sufficiently below the output LOW level of V_{CC} – 1.8 V*

Figure 4.8 DC Coupled TAXI Fiber Optic Interface (Terminated)



17490A-16

Note:

1. *If the optical and TAXI power and ground planes are decoupled as described in Section 5.2, AC coupling is always recommended to allow for variations in power and ground plane voltages. AC coupling is discussed in Section 4.1.4, Section 4.1.8 and Section 4.3.*

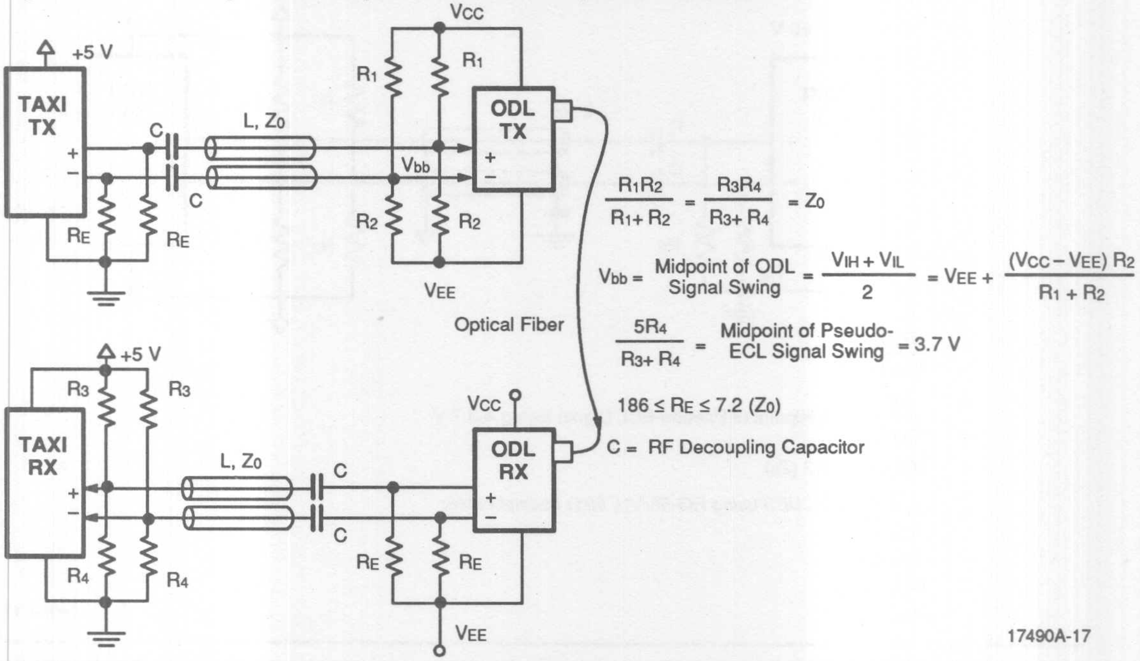
4.1.8 AC Coupled TAXI Optical Fiber Interface

If the TAXI-275 set and optical components operate on different power supplies, then the TAXI-275 optical interconnect requires AC coupling.

The recommended configuration for an AC capacitor coupled TAXI-275 optical components interconnection is shown in Figure 4.9. In this configuration, R_E is the ECL output pull-down resistor, C is the AC coupling capacitor, L is the length of the transmission line, Z_0 is the characteristic impedance of the coax, twisted pair or microstrip, and R1 and R2 are a matched line termination and a voltage divider.

The configuration in Figure 4.9 is recommended for any line length, L, which separates the TAXI and the optical module by greater than 1.5 inches. The V_{bb} bias voltage determined by the parallel terminations is always needed for AC coupled links.

Figure 4.9 AC Capacitor Coupled TAXI-275 to Optical Transceiver Interface


VCC
+5 V
TAXI TX
+
–
C
C
RE
RE
L, Z0
R1
R1
Vbb
R2
R2
VEE
ODL TX
+
–
Optical Fiber
+5 V
R3
R3
TAXI RX
+
–
R4
R4
L, Z0
C
C
RE
RE
VCC
ODL RX
+
–
VEE
$\frac{R_1R_2}{R_1+R_2} = \frac{R_3R_4}{R_3+R_4} = Z_0$
V_{bb} = Midpoint of ODL Signal Swing = $\frac{V_{IH}+V_{IL}}{2} = V_{EE} + \frac{(V_{CC}-V_{EE})R_2}{R_1+R_2}$
$\frac{5R_4}{R_3+R_4}$ = Midpoint of Pseudo-ECL Signal Swing = 3.7 V
$186 \leq R_E \leq 7.2\ (Z_0)$
C = RF Decoupling Capacitor
17490A-17

4.1.9 Interfacing to Coaxial Cable

In many applications, system cost can be reduced without sacrifice in performance by using coaxial cable as the serial media. Unlike optical fiber, which requires optical components between the fiber and the TAXI-275 set, coaxial cable can be connected directly to the TAXI TX/TY serial output pins. Optical components do not have to be purchased resulting in cost reduction.

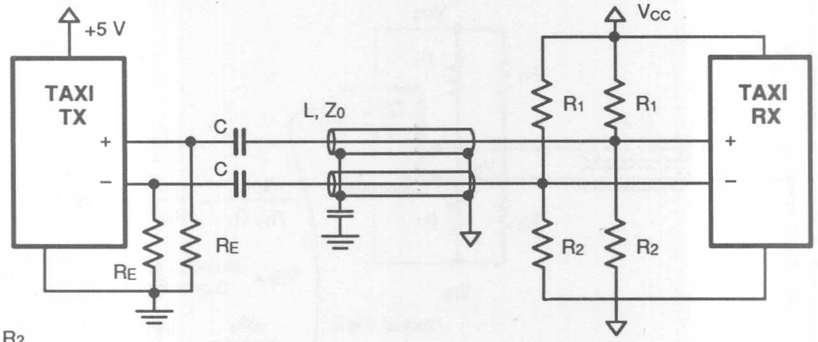
Coaxial cable can be used as serial medium for short-to-moderate length links. At longer lengths, the advantages of fiber optic transmission (low attenuation, immunity to EMI and ground loops) make it the media of choice.

The maximum length possible for a coaxial cable TAXI link depends on the type of coaxial cable used and the data rate. Higher data rates will tend to reduce coaxial link lengths because attenuation and pulse dispersion increase with frequency causing higher degradation of the signal. Many different types of coaxial cables are available with various degrees of attenuation. However, low attenuation coaxial cable generally means an increase in cable size and rigidity.

Interfacing the TAXI-275 set with coaxial cable as the serial media is quite simple. Appropriate line terminations are required, and either capacitor or transformer coupling can be used. The recommended configuration, including the necessary design equations for capacitor coupling, is shown in Figure 4.10 (see section 4.3 for transformer coupling). Each of the components that make up the interface serve the same purpose as in the AC coupled TAXI-275 to optical transceiver interface shown in Figure 4.9.

Note that two coaxial cables comprise the link, one for each of the differential pseudo-ECL signals. These two lines should be electrically tested to ensure an electrical length difference of less than 0.2 ns.

Figure 4.10 Coaxial Cable Interface



$$\frac{R_1R_2}{R_1+R_2} = Z_0$$

$$\frac{5R_2}{R_1+R_2} = \text{Midpoint of Pseudo-ECL Signal Swing} = 3.7\text{ V}$$

$186 \leq R_E \leq 7.2\ (Z_0)$

SAMPLE VALUES using RG-58A/U, 50 Ω coaxial cable:

$R1=68\ \Omega$

$R2=200\ \Omega$

$R_E=300\ \Omega$

$C=0.1\ \mu F$

17490A-18

4.2 INTERFACING TO TWISTED-PAIR CABLE

Another copper interface alternative is twisted pair cable. Twisted pair cable is generally more lossy than coaxial cable making it suitable only for shorter transmission distances.

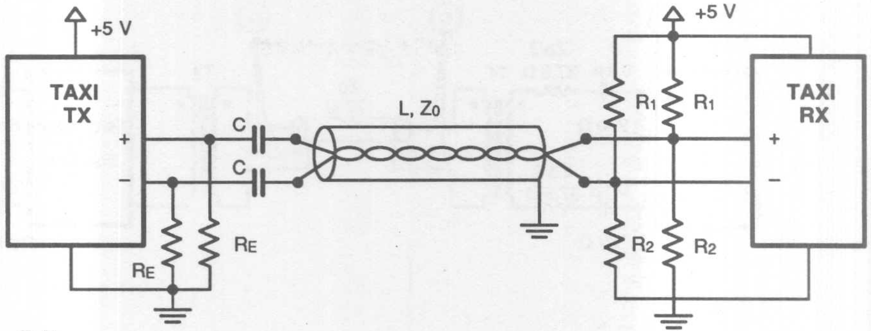
To reduce the possibility of noise being induced along the line, twisted-pair cable with shielding is recommended. Type 1 Shielded Twisted Pair (STP) cable with a Z_0 of 150 Ω is recommended for use at the speeds of the TAXI-275.

Transformer coupling is recommended for twisted pair media as shown in Section 4.3.

Using the TAXI-275 set with shielded-twisted-pair as the serial media is similar to using it with coaxial cable. One possible configuration is shown in Figure 4.11, where each of the components that make up the interface serve the same purpose as in the AC coupled TAXI-275 fiber optic interface shown in Figure 4.9. The new values for the150 Ω STP were calculated using the formulas in Section 4.1.2 through Section 4.1.4.

Note that with shielded-twisted-pair, only one cable is required to form the link. The twisted-pair conductors carry the differential pseudo-ECL signals and the shield is grounded at the Receiver. The Type 1 cable consists of two pairs that can be used to form a full duplex link.

Figure 4.11 Shielded-Twisted Pair Cable Interface



$$\frac{R_1R_2}{R_1+R_2} = Z_0/2$$

$$\frac{5R_2}{R_1+R_2} = \text{Midpoint of Pseudo ECL Signal Swing} = 3.7\text{ V}$$

$186 \le R_E \le 7.2\ (Z_0)$

C = RF Capacitor

SAMPLE VALUES: Using Type 1 150 Ω shielded twisted pair cable, a successful TAXI link was established using the following component values:

R1=100 Ω

R2=284 Ω

R_E=300 Ω

C=0.1 µF

17490A-19

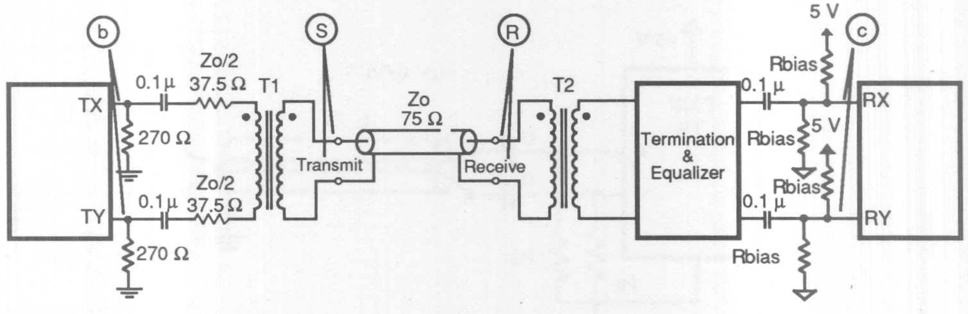
4.3 AC TRANSFORMER COUPLED

Transformer coupling isolates the Transmitter and Receiver DC bias from the transmission line. The transmission media is therefore floating without a DC reference.

Transformer coupling, designed as shown in Figure 4.12 below, filters out common mode noise and allows for longer distances and better noise immunity.

Transformer coupling can be used with coaxial or twisted pair cables. Figure 4.12 illustrates a 75 Ω coax circuit. The points specified as b, S, R and c are points where the transmission signals are specified in the Fibre Channel standard. The compensation or equalizer circuit is illustrated in the following section.

Figure 4.12 Transformer Coupling of a 75 Ω Coaxial Cable

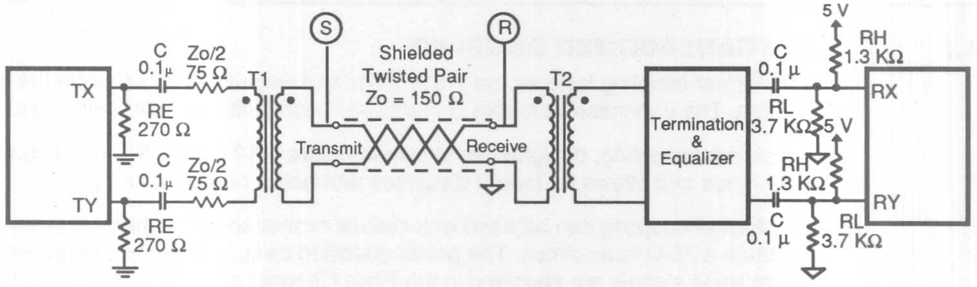


17490A-20

The twisted pair transformer coupling circuit is very similar to the coax circuit. Normally, twisted pair media tend to limit the distance of operation of a link more than coaxial media. Figure 4.13 illustrates the component values for a STP cable with Z_0 equal to 150 Ω.

The comparative cost of transformer coupled interfaces and capacitor coupling interfaces is potentially insignificant. The transformer requires additional board space. Some of the determining factors in making a decision to use transformer coupling should be the transmission distance, noise immunity, board space and cost requirements of the system.

Figure 4.13 Transformer Coupling of 150 Ω STP



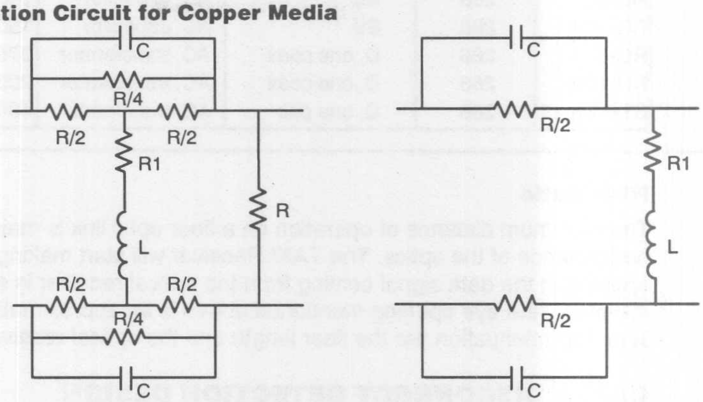
17490A-21

The components used in transformer coupling are of standard values and tolerances The resistor tolerance is 1% and the capacitor tolerance is 10%. Transformers optimized for Fibre Channel frequency range or other ranges can be obtained from a number of vendors—see Appendix C.

4.4 COMPENSATION

The Equalizer in Figure 4.12 and 4.13 contains some form of compensation. Compensation is used to counteract the differences in amplitude loss and propagation delay between frequency components of the transmitted signal. Compensating a link can increase the link distance by about 20 to 30 percent over the same but noncompensated link. The method of compensation shown on the left in figure 4.14 below is suggested in the Fibre Channel FC-0 specification. The method of compensation on the right is a simpler method and is almost as efficient.

Figure 4.14 Compensation Circuit for Copper Media



R = Zo of cable. R1, L, C should be chosen based on cable used and data rate.

4.5 LINK PERFORMANCE

4.5.1 Jitter

Jitter tolerance preliminary information should be available in Q1 of 1993.

4.5.2 Distance/Failure Knee

Copper

Preliminary tests have been performed in order to gather data for the Fibre Channel working group.

The test set-up consists of:

- Pattern Generator—HP16500 system
- TAXI-275 copper link using TAXI-275 Evaluation Board
- Logic Analyzer—HP16500 system

A pseudo random data pattern programmed into the Pattern Generator for the transmitter side is transmitted along the link and tested for errors by the Logic Analyzer on the receiving side. The maximum distance is the length of the cable at which there are no errors for at least 16 hours of operation. This means a BER<10^{-12} @ 95%. The cable is normally kept on the bench, which is probably a less noisy (than real life) situation. On the other hand, the transceiver board is not protected in any way (shields, etc.) from the outside noise.

Test conditions:

- No equalization.
- Cable is terminated at both ends: serial termination for the transmitter and parallel for the receiver; 75 Ω for RG-6 and RG-179, and 150 Ω for STP Type 1.
- The eye opening that seems to be the minimum required for the error rate has a width of around 60 mV pp and 1 ns pp.

Table 4–1 **Copper Media Test Results**

Cable Type	Baud Rate, Mbaud	Single Ended or Differential	T and R Coupling	Measured Distance	Fibre Channel Specification
RG-6	266	SE	AC, capacitor	300 feet	
RG-179	266	SE	AC, capacitor	150 feet	
RG-6	266	D, one coax	AC, transformer	375 feet	75 meters
RG-179	266	D, one coax	AC, transformer	200 feet	30 meters
STP-Type1	266	D, one pair	AC, transformer	100 meters	50 meters

Fiber Optic

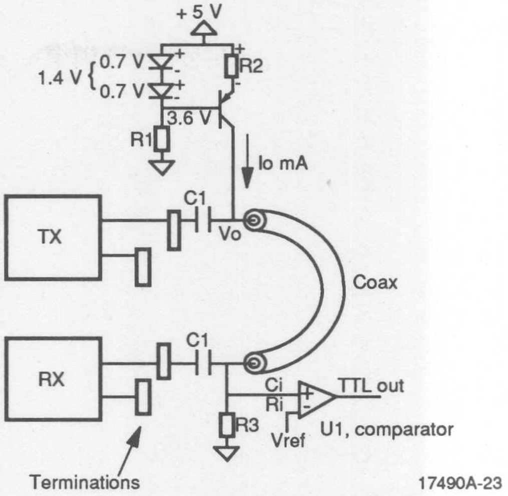
The maximum distance of operation on a fiber optic link is mainly determined by the performance of the optics. The TAXI Receiver will start making errors when the eye opening in the data signal coming from the optical receiver is smaller than the required minimum (the eye opening mentioned above is an approximation). The main sources of jitter and attenuation are the fiber length and the optical receiver frequency response.

4.6 CABLE DISCONNECT DETECTION DESIGN

It may be necessary to disable the logic that processes the Receiver outputs when a cable disconnection has occurred. When the cable connection to the Receiver serial inputs is broken, the Receiver inputs return to the voltage determined by the values of the terminating resistors. Normally, the disconnect situation will produce a constant violation state on the outputs of the Receiver.

If the Receiver serial input termination values are very close to each other, a very small difference in termination voltage between the Rx voltage level and the Ry voltage level is produced. Under these conditions, it is possible for noise in the system to produce transitions that the Receiver will try to interpret as data. When this occurs, violations and random data and commands could be output by the Receiver. It might be difficult to disable the processing of the invalid data being received with software or hardware techniques using the Receiver outputs. In order to avoid this situation, the hardware circuit Figure 4.15 can be used.

Figure 4.15 Receiver Side Logic for Detection of Cable Disconnect



The cable disconnect detection circuit is designed to produce a TTL signal indicating to the Receiver logic that the cable is disconnected (Low) or connected (High). The two capacitors C1 in the serial stream isolate the serial signals from any DC level offset to prevent loading of the serial signal by the added detection logic—the serial cable is floating at a different voltage level. The resistor R3 in parallel with the input resistance (Ri) to GND of the comparator U1 connect the isolated serial signals to a DC level of a determinate voltage based on the output current Io. The transistor, diode and resistor network at the Transmitter end of the cable act as a current source setting the DC voltage level of the serial link to a value determined by R3.

The serial signals in the cable will be at zero volts when the cable is disconnected creating a voltage level below V_{ref}, producing a TTL Low output. The cable disconnection process will more than likely produce several disconnected/connected states before stabilizing. Hysterisis or capacitive filtering can be used to debounce the comparator outputs.

The devices used in the circuit, except for the comparator, follow the recommended tolerances for TAXI products. The comparator has the requirement that the input capacitance Ci must be 10 pF or less. If Ci is larger than the recommended value the serial signal edge rates will be significantly affected causing a reduction in distance performance. A low capacitance value is more typical of faster devices.

The time it takes for the detection circuit to detect a stable cable disconnection is more than likely going to be in the millisecond range. The mechanical disconnection will be the main determining factor. The electrical section of the detection circuit is dependent upon the delay through the comparator and its debouncing circuitry.

CHAPTER

5 BOARD LAYOUT CONSIDERATIONS

While the TAXI-275 devices are digital in application, they do contain analog circuits which contain the high frequency analog Phase Locked Loops. Reliable operation in a high frequency analog and digital environment requires that certain board layout rules be followed.

For example, TAXI-275 applications which are laid out on a wire wrap board will not work reliably. Most wire-wrap cards have insufficient separation between small signal current and digital switching current because they have at most one power and ground plane for all the logic and because the signal lengths are longer. Digital switching noise can couple into the analog PLL, causing phase errors and loss of synchronization. The preferred realization of a TAXI application is on a printed circuit board, where the user can control the layout of power and ground planes and signal lines.

Additional general information on high speed layout techniques is available as an application note—High-Speed-Board Design Techniques, PID No. 16356A, issued February 1992.

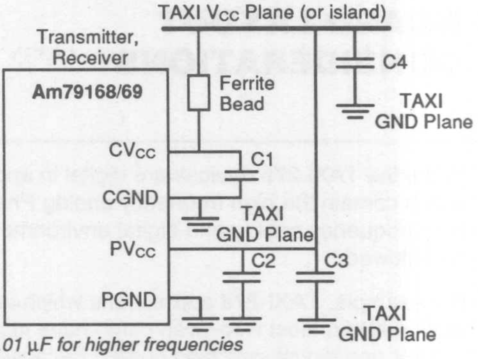
5.1 PRINTED CIRCUIT BOARD LAYOUT

5.1.1 Rules for Layout

The following rules should be followed to ensure minimal noise coupling:

- Use a PC board with separate GND and V_{CC} planes (or islands) for the TAXI-275 devices, see Figure 5.4. There should not be any signal lines from other sources passing under the TAXI power islands.
- Use two to three capacitors which differ by at least a factor of ten in value to decouple the TAXI-275 devices. The reactance of large capacitors has a significant inductive component at high frequencies. Because of this inductive component, a single large capacitor is not very effective against high frequency noise. Two capacitors, one typically of 0.1 µF and one of 0.01 µF are more efficient at de-coupling than a single large capacitor of equivalent capacitance. It may also be beneficial to use a third capacitor of 1 µF. The recommended decoupling is shown in Figure 5.1.

Figure 5.1 Transmitter and Receiver Decoupling Schematics



C1 = 0.1 µF (ceramic) or optional 0.01 µF for higher frequencies

C2 = 0.1 µF (ceramic)

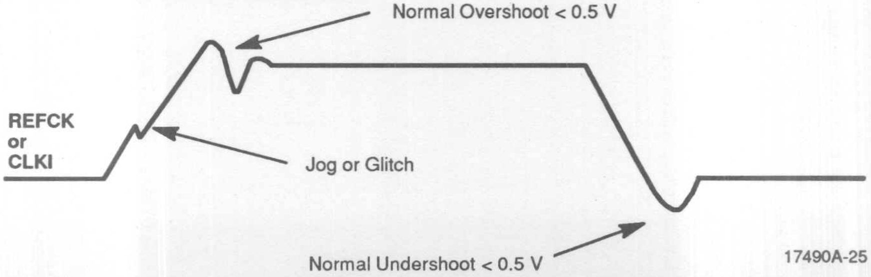
C3 = optional 1 µF Tantalum

C4 = 10 µF (Tantalum) for use at junction between TAXI-275 power planes and logic power planes

17490A-24

- Keep all bypass capacitors as close to the power pins of the device as possible. The CV_{CC}/CGND bypass capacitor should be connected to the TAXI-275 CV_{CC} and CGND pins through heavy traces. The CV_{CC} pin should be connected to the TAXI-275 V_{CC} plane through a Ferrite Bead (see Appendix C.4 for examples). The CGND pin should be connected directly to the GND plane. The PV_{CC}/PGND capacitor pins should be connected directly to the V_{CC} and GND planes. The distance between the capacitor pins and the TAXI-275 power pins should be minimized, preferably less than a 1/4 inch. On surface mount boards, avoid the use of microbarrel feed throughs for connecting V_{CC}/GND pins to the V_{CC}/GND planes. If feed throughs must be used, multiple feed throughs should be used for each connection.
- Use a high quality RF grade capacitor. Type COG and NPO capacitors are well suited for the above application. X7R capacitors can also be used. The use of Z5U capacitors is not recommended. Refer to Appendix C for more information on capacitors.
- Ensure that the power supply does not have more that 100 mV of peak-to-peak noise at any of the TAXI-275 power pins. Make this check while the TAXI-275 devices are sending random data.
- Care should be taken to ensure that no jogs or glitches occur in the CLKI or REFCK signal as shown in Figure 5.2. If present, these glitches will be passed onto the PLL and cause an occasional error.

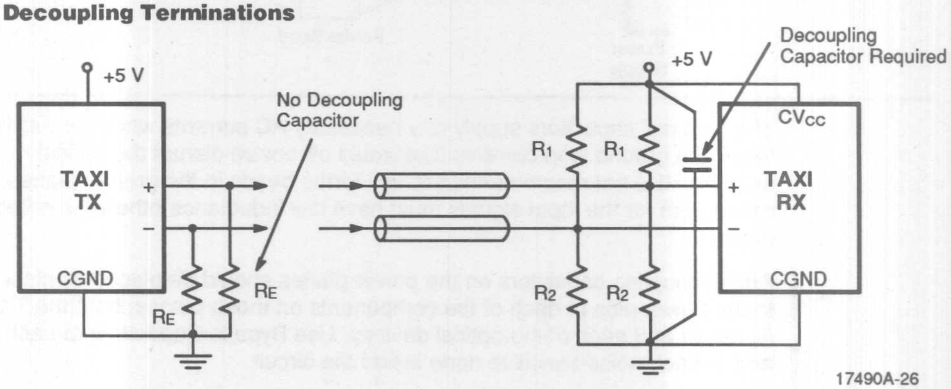
Figure 5.2 Jogs and Glitches in the Clock Line



17490A-25

- Run the serial outputs parallel to each other, or one on top of the other at all times and route them away from the Transmitter. Do the same for serial inputs on the Receiver. Running the serial traces adjacently will minimize noise caused by these extremely fast signals on other traces. Use of strip lines for serial signals is recommended.
- When terminating serial lines to or from the TAXI-275 ensure that the V_{CC} rail or ground tap is not at a noisy location. Resistors can couple noise from a power supply rail into the serial lines. V_{CC} to GND decoupling adjacent to the resistors is recommended when using pull-up/pull-down terminating resistor setups as shown in Figure 5.3. When using only a pull-down, do not use decoupling.

Figure 5.3 Decoupling Terminations



Note:

1. *It is not recommended to use sockets in production product. Most sockets available tend to lower the high frequency performance of the TAXI link.*

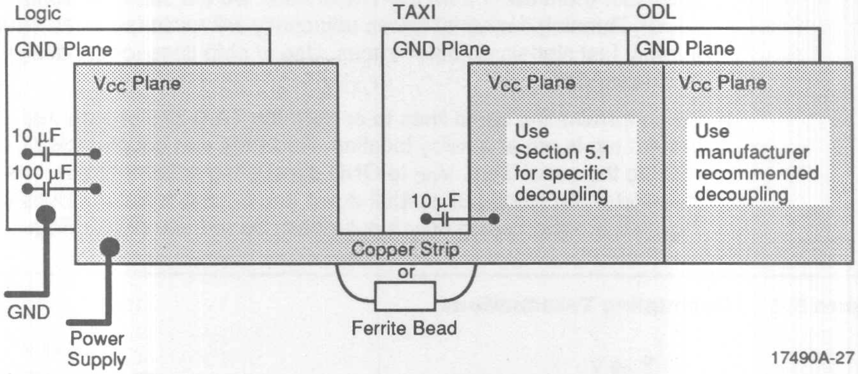
5.2 LAYOUT USING FIBER OPTIC DATA LINKS

Because of their small signal levels, fiber optic data links require some care in layout. Fiber optic data Link receivers consist of a photo sensitive diode and an amplifier. The photo-diode converts light pulses into currents of around a few hundred nano-amps. This signal current is then amplified and translated into an ECL signal.

TAXI Receivers and most digital chips switch hundreds of milliamps. If switching noise from the digital section of the board gets coupled into the optical data link, the signal from the light pulse data can be corrupted. To prevent the coupling of the optical data link output with other digital signals, the user must ensure that small signal and digital switching currents do not flow in the same path. Usually, the ODL manufacturer will recommend a network to filter the power inputs to the device. using a section of the TAXI power planes as illustrated in Figure 5.5 for the ODL plane with the manufacturer recommended power coupling should be all that is necessary for proper layout.

Guaranteeing the uniqueness of current flow in these signal paths can be done by separating both the optical V_{CC} plane and the optical GND plane from the V_{CC} and GND planes used by other digital circuitry including the TAXI devices—the dividing line in Figure 5.4 illustrates the separation. Either a copper strip or a ferrite bead can be used to separate the ODL plane from the other logic. Separating the planes is only necessary in circumstances where the expected noise level is high.

Figure 5.4 Fiber Optic Data Link Decoupling


Logic
TAXI
ODL
GND Plane
GND Plane
GND Plane
VCC Plane
VCC Plane
VCC Plane
10 µF
100 µF
Use Section 5.1 for specific decoupling
Use manufacturer recommended decoupling
10 µF
Copper Strip or Ferrite Bead
GND
Power Supply
17490A-27

The 'bypass' capacitors supply any necessary AC currents while the Ferrite Beads break the ground loop currents that would otherwise disrupt the optical receiver's amplifier. It is not recommended to use ferrite beads in the ground planes. The GND return path for the input signals must have low inductance otherwise reflections can occur.

Any decoupling capacitors on the power planes should be placed as close as possible to the power pins of each of the components on those planes; both the Transmitter and Receiver and each of the optical devices. Use Bypass Capacitors at each noise source and at each noise sensitive node inside the circuit.

CHAPTER

6 INTERFACING TO WIDE BUSSES

6.1 WIDE BUS INTERFACING WITH THE TRANSMITTER

6.1.1 Introduction

The TAXI-275 Transmitter/Receiver pair provides a cost effective solution for transmitting data of 8-bits or wider over long distances. The Transmitter section will be described in the following and the Receiver section will be described in Section 6.2 of the Manual. The following example design economically multiplexes 32-bit data words using one Am79168-275 Transmitter. The design note can easily be applicable to systems with wider or narrower data paths with or without commands.

6.1.2 Advantages

There are several advantages to using the multiplexed data scheme utilizing one TAXI-275 Transmitter as opposed to a system using several Transmitters or ribbon cable:

- To implement the multiplexer circuit for 32 bits requires one Am79168-275 Transmitter and four SSI integrated circuits. A 32-bit wide data path without multiplexing requires four Am79168-275 Transmitters.
- Four Am79168-275 Transmitters require more board real estate than four SSI parts and one TAXI-275 Transmitter.
- Four Am79168-275 Transmitters will typically dissipate 2.8 W, while one Am79168-275 and four SSI chips typically dissipate 1 W. The power savings is even more dramatic if optical data links are being used.
- The cost of coaxial cable is considerably less than the same length of ribbon cable.

6.1.3 Implementation

Implementation of the 32-bit multiplexed Transmitter circuit is shown in Figure 6.1. In addition to the Am79168-275 Transmitter, the following parts are required:

(1) 74LS00	quadruple 2-input positive NAND gates
(1) 74LS04	hex inverters
(1) 74LS20	dual 4-input positive NAND gates
(1) 74LS174	hex D-type flip-flops

Additionally, Tri-state registers capable of capturing the 32-bit word are also required (Am29C821s in this example). These registers should already be available in the host system.

Options

The design can be optimized to save board space by fitting it into a PAL16R6 device or equivalent.

6.1.4 Operation

Referring to Figure 6.1 Logic Diagram of 32-bit Multiplexed Transmitter, the data is assumed to be simultaneously loaded into the registers when the STROBE signal is input to the system. Multiplexing is provided by a D flip-flop shift register whereby a "zero" is shifted through the D flip-flops enabling one transmit register at a time. The STROBE pulse is used to initiate the shift register process. The NAND gate (U1) at the input of D1, ensures that only a single "0" is possible while the registers are being selected. The TAXI-275 CLKI signal used to clock the shift register is inverted to provide adequate set-up time to ensure that no false strobes reach the TAXI-275. The four-input NAND gate (U2) and two-input NAND gate (U3) are used to ensure that the TAXI-275 will be strobed while there is data available in the registers.

The Jumpers are provided to allow selection between ACK0 and ACK1 modes. If J2 is shorted, the system will run in "AUTO REPEAT ACK0" mode. If J3 is shorted, the system will run in "AUTO REPEAT ACK1" mode. This means that there will be a STRBI on every clock cycle. If the output of ACK1 is shorted back to the strobe input, the system will run in "auto-repeat ACK 1" mode. This means that a SYNC will be automatically inserted after each 32-bit word. Figure 6.2 shows the timing for the strobe initiated mode. Figure 6.3 shows the timing for auto repeat ACK0 and auto repeat ACK1 modes.

Figure 6.1 Logic Diagram of 32-Bit Multiplexed Transmitter

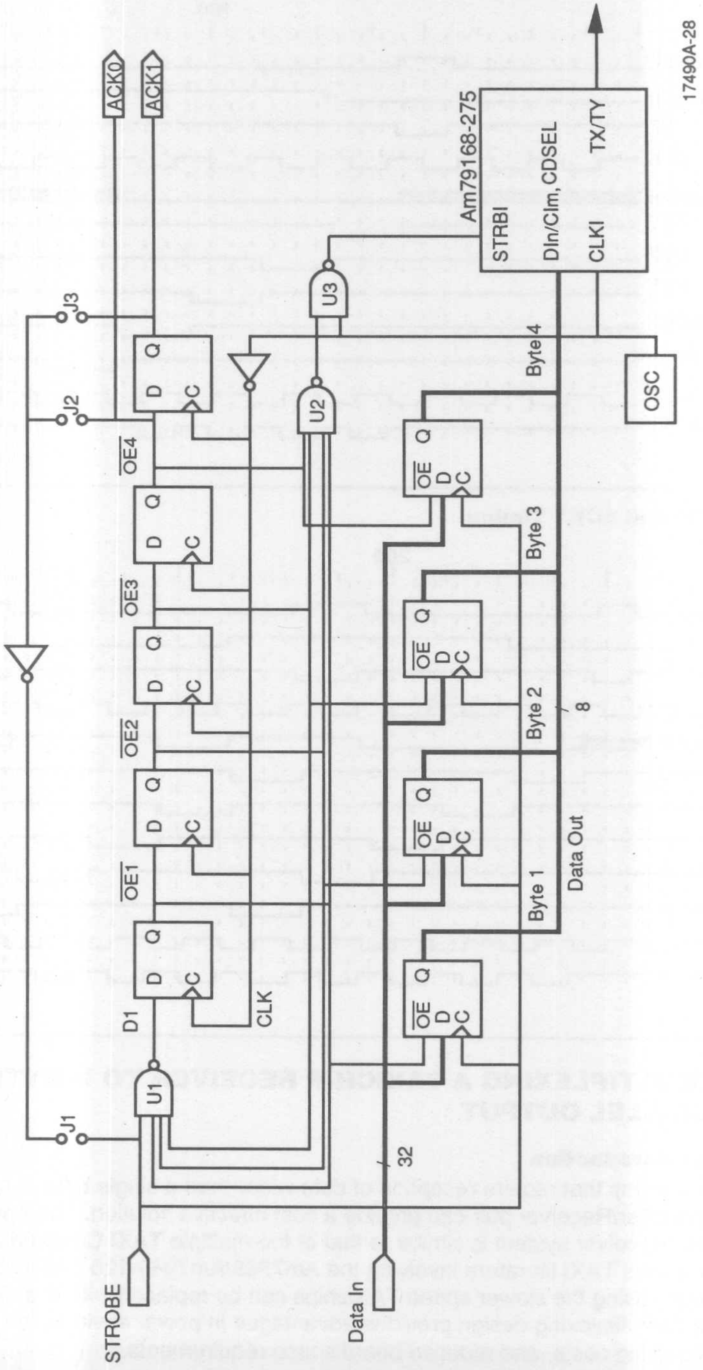
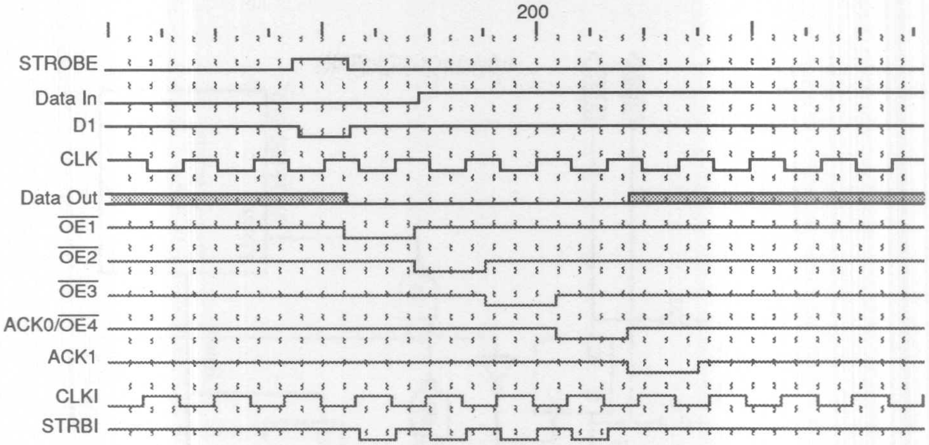
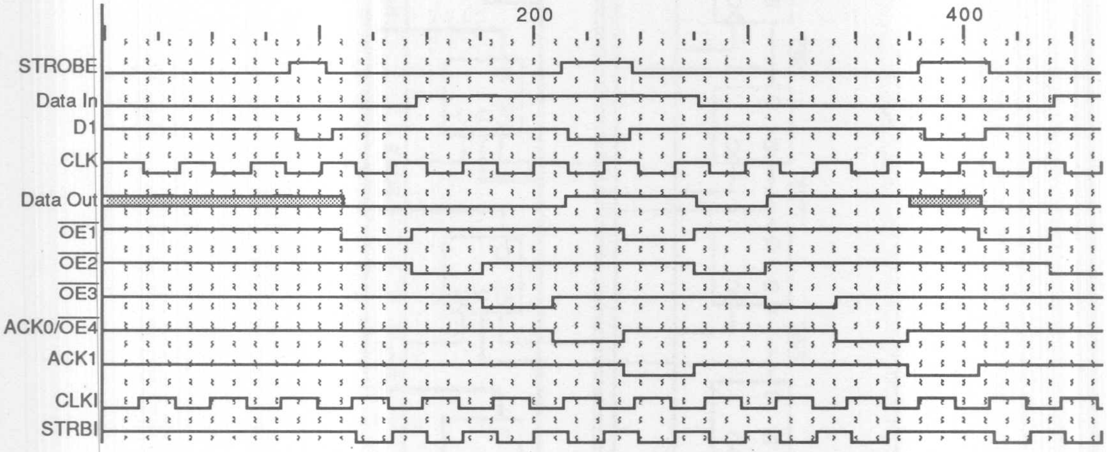


Figure 6.2 Strobe Initiated Timing For 32-Bit Transmitter Interface



17490A-29

Figure 6.3 ACK0 and ACK1 Timing



17490A-30

6.2 DEMULTIPLEXING A TAXICHIP RECEIVER TO N-BYTE PARALLEL OUTPUT

6.2.1 Introduction

For systems that require reception of data wider than a single byte, a single TAXI-275 Transmitter/Receiver pair can provide a cost effective solution. The operation of the single Receiver system is similar to that of the multiple TAXI Cascade system described in previous TAXI literature involving the Am7968/Am7969-125 TAXIchips. Parallel link designs using the slower speed TAXIchips can be replaced with this method as well. This demultiplexing design provides advantages in power savings, lower component and cabling costs, and reduced board space requirements.

An example design is used to illustrate the basic concepts involved in demultiplexing data from the TAXI-275 receiver. The particular example design demultiplexes four bytes of data/commands into one 32-bit word. The 32-bit word can consist of mixed 8-bit data bytes and 4-bit command bytes. Figure 6.4 illustrates the example logic design using nominal commercial devices and Figure 6.5 shows the associated circuit timing.

6.2.2 Functional Description For Example Design

The example design uses the Receiver outputs to generate the needed signals to demultiplex four consecutive bytes of data into one parallel 32-bit word. The design is divided into three basic sections: (1) the Data Path and Control Logic, (2) the Data Capture Logic and (3) the External Data Path Logic. Data and command indication as well as violation status are indicated for each byte using two extra bits—D/$\overline{\text{CMD}}$ and VLTN.

Data Path and Control Logic

From the Receiver outputs the data flows through a pair of multiplexing buffers to the four sequentially clocked data registers and then to the external data destination. The multiplexing buffers, controlled by S1 and S2 receiver outputs, are used to select between the Command line outputs and the Data line outputs. When S1 is High and S2 is Low the Command lines are selected using the D/$\overline{\text{CMD}}$ signal. When S1 and S2 are Low the Data lines are selected using the $\overline{\text{DATA}}$ signal.

The byte information and status are generated from the S1 and S2 outputs as well. Every byte of the four bytes that form the 32-bit word has a violation bit and a command/ data indication bit accompanying it. The violation bit (VLTN) is generated when S1 is Low and S2 is High. The command/data bit (D/$\overline{\text{CMD}}$) is Low for Commands and High for Data as indicated when S1 is High and S2 is Low and S1 and S2 are Low, respectively.

Data Capture Logic

The Data Capture Logic consists of a shift register constructed of four D flip-flops and a 3-input NOR gate. This logic generates the sequential clock pulses that capture the data/command bytes to form the 32-bit word required by the external data path. The shifter is loaded with a '1' that propagates through the flip-flops sequentially clocking the first column of four registers which capture the incoming data. When the '1' is shifted through the fourth flip-flop, it raises the CLK4 signal which is involved in the generation of the PCO signal. The PCO signal is described in the External Data System Logic section.

The main clock used for the data capture logic is the STRBO signal. The CLKO signal along with the SYNC_RCVD signal are used to generate the reset of the D flip-flops. The STRBO signal rises a few bit times before the CLKO signal and is therefore used to allow the first K28.5 Command or SYNC symbol to be output to the external system before CLR_CNTR resets the data capture logic. The purpose for outputting the SYNC symbol to the external system is to indicate a short data word—one with less than four bytes.

The CLR_CNTR signal is generated by the AND of the CLKO signal and the state of the S1, S2 and C2–C0 Command lines. A Low level indicates that a SYNC symbol is present. Only C2–C0 are used in the logic because the Receiver indicates a Command 5 or Command 13 on the Command lines when a SYNC symbol is received depending on the Running Disparity (RD). Note that C3 can be used as a RD indicator when receiving a SYNC symbol.

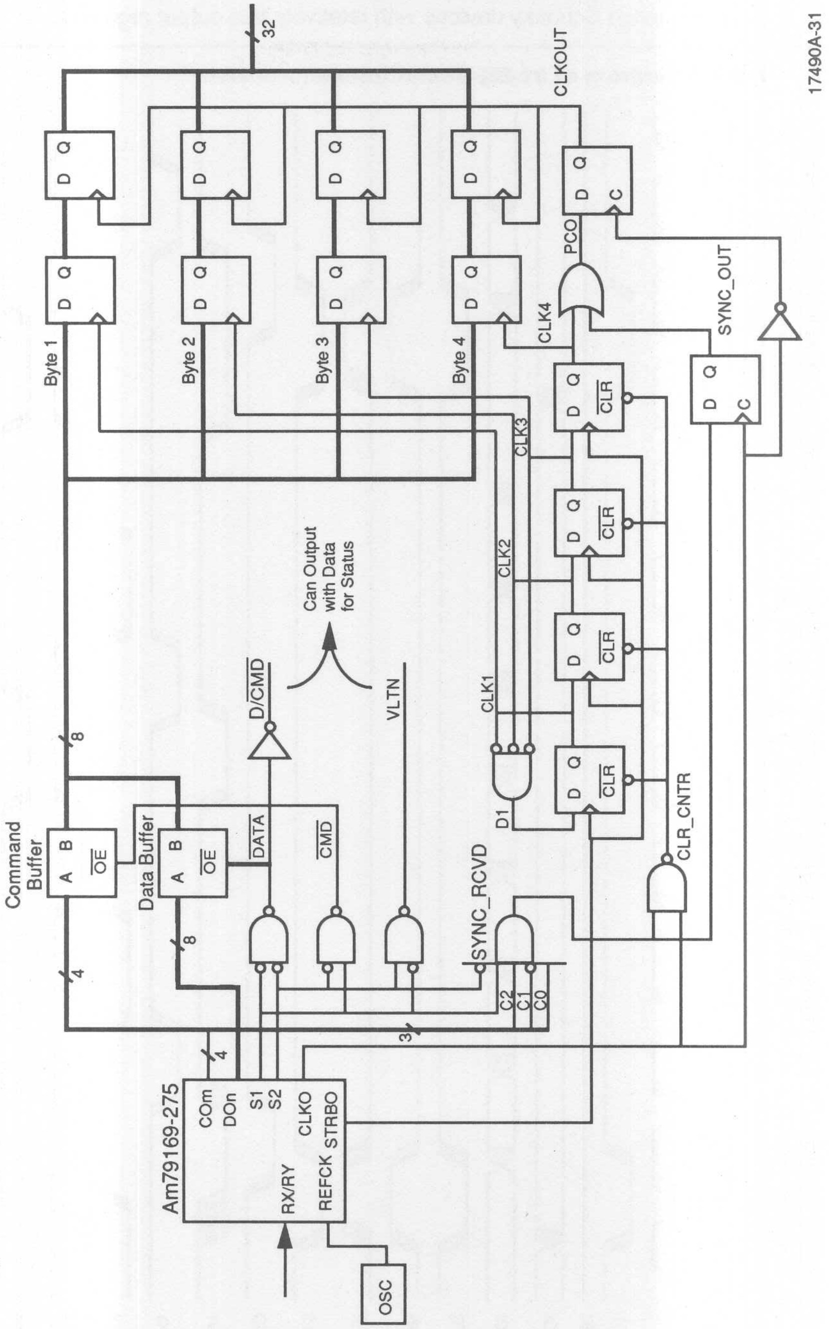
SYNC symbols are used to keep the TAXI-275 Receiver locked onto the transmitted byte rate and byte boundary. A pair of SYNC symbols will also reset the Data Capture Logic to the proper word alignment. SYNC symbols are automatically sent by the Transmitter whenever a byte time passes without a STRBI (no data to send) pulse.

External Data System Logic

The 32-bit data word is clocked out to the external system by the CLKOUT signal. The CLKOUT signal is generated by clocking PCO into a D flip-flop with the $\overline{\text{CLKO}}$ signal. The PCO signal is generated from the OR of the CLK4 signal and the SYNC_OUT signal. The SYNC_OUT signal is generated by clocking the SYNC symbol indicator SYNC_RCVD into a D flip-flop with the CLKO signal. The SYNC_RCVD indicator is generated from the Command lines C2–C0 and the status lines S1 and S2 when a Command 5 or Command 13 is received.

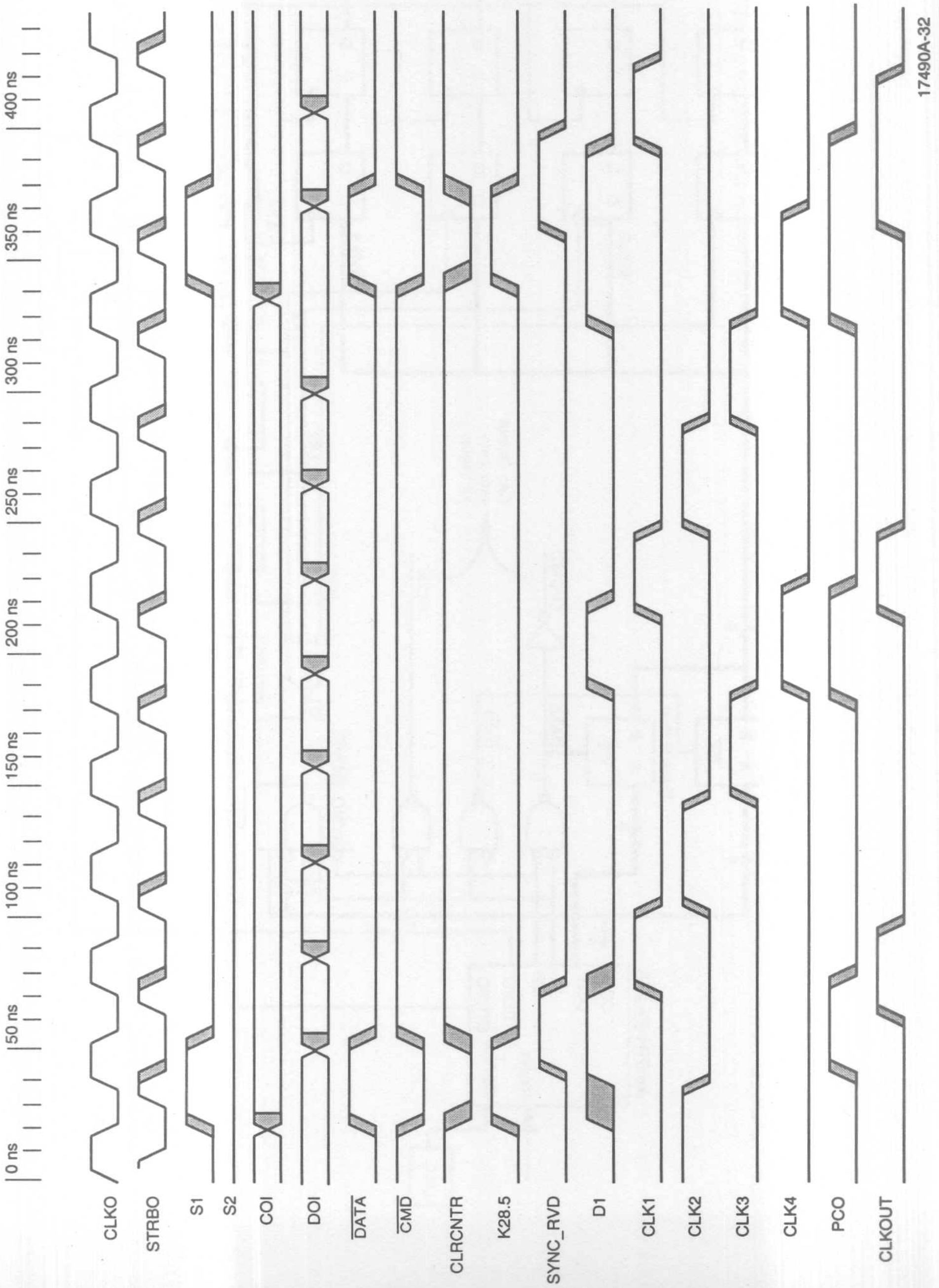
The CLKOUT signal occurs under three circumstances. The first occurrence is when Data alone is being correctly received. Under this condition, the fourth byte being received will generate a CLK4 signal which will strobe CLKOUT on the next rising edge of $\overline{\text{CLKO}}$. Another occurrence is when four bytes are being received with a SYNC symbol(s) used as a word delineator. Under these conditions, the first SYNC symbol between words will clock out the Data. The following SYNC symbols will not change the state of the data. The last occurrence is when a 32-bit word is shorter than expected. This could occur if data is corrupted into an alias SYNC symbol or if the Transmitter is strobed in the t_6 zone where SYNC characters will occur randomly. A CLKOUT signal before the word is completed would indicate an error under these circumstances.

Figure 6.4 Logic Diagram of 32-Bit Demultiplexed Receiver



This design should work with any high performance logic, although logic families should not be mixed unless timing considerations have been made. This particular example uses Advanced Schottky devices with relatively fast output registers.

Figure 6.5 Timing Diagram of 32-Bit Demultiplexed Receiver



6.2.3 Timing Considerations

To prevent glitches on clock signal and the potential capture of incorrect data, the timing between STRBO rising and CLR_CNTR becoming active must be considered. The timing diagram is shown in Figure 6.5.

An early SYNC character in the 3rd byte position (roughly 20 ns to 60 ns on the scale), is followed by a normal data repeat sequence from 60 ns to 340 ns. A SYNC character follows as the word separator. The device tolerances are shown by the shaded regions.

CHAPTER

7 UPGRADING FROM Am7968/69 TO Am79168/Am79169-275

7.1 INTRODUCTION

The following material is provided to ease the process of upgrading from the Am7968/9 devices to the new high-speed Am79168/Am79169-275 devices. A description of pinout and signal functionality differences is presented that should be considered when redesigning an existing board. The layout and decoupling recommendations remain the same as those recommended in the Board Layout Recommendations section of this Technical Manual. For details of the exact functionality of any device mentioned in the following report, please refer to the appropriate data sheet.

7.2 BASIC FEATURES OF THE Am79168/9-275 DEVICES

The new Am79168/9-275 TAXIchips have several enhancements over the older Am7968/9 family of devices. The Am79168/Am79169-275 devices have an increased speed of operation which runs between 175–275 Mbaud on the serial link. The Am79168/Am79169-275 devices use 8B10B encoding which is more robust than 4B5B coding in its capability of detecting errors. The DC balance attribute of the 8B10B encoding scheme eliminates DC shift problems that may occur with an AC coupled system. The Am79168/9-275 TAXIchips are available in a PLCC package only.

7.3 SPECIFIC FUNCTIONALITY DIFFERENCES

Table 7.1 Transmitter—Am7968 vs. Am79168-275

Differences in Pinouts and Related Functionality (Pinouts for PLCC Packages)

Pin #	Am7968	Am79168	Comments
1	ACK	DI6	
2	STRB	DI7	
3	SEROUT+	CI3/DI8	
4	SEROUT-	CI2/DI9	
5	VCC2	CI1	
6	VCC1	CI0	
7	VCC3	CDSEL	CDSEL Command or Data selection, strobed in signal
8	$\overline{\text{RESET}}$	CVCC	$\overline{\text{RESET}}$ no longer needed
9	DMS	STRBI	
10	CLS	ACK	CLS no longer needed, no on-chip cascading with Am79168
11	SERIN	$\overline{\text{OE1}}$	SERIN no longer used, /OE1 serial output TX enable
12	CI0	$\overline{\text{OE2}}$	$\overline{\text{OE2}}$ serial output TLOOP enable
13	CI1	TEST	TEST factory test pin
14	CI2/DI9	TX	TX equivalent to SEROUT+
15	CI3/DI8	PVCC	
16	DI7	TY	TY equivalent to SEROUT-
17	DI6	DMS	DMS with 8-bit and 10-bit word selection only
18	CLK	TLOOP	CLK no longer needed, TLOOP serial loopback test output
19	X2	$\overline{\text{ESEL}}$	$\overline{\text{ESEL}}$ encoder enable, a strobed in signal
20	X1	CLKI	CLKI a TTL clock source only so X2 for crystal not needed
21	GND2	PGND	
22	GND1	CGND	
23	DI0	DI0	
24	DI1	DI1	
25	DI2	DI2	
26	DI3	DI3	
27	DI4	DI4	
28	DI5	DI5	

Signal Group Differences

Data Signals: DI7–DI0:		Am7968 pin # 16,17,28–23 respectively		
		Am79168 pin # 2,1,28–23 respectively		
		no functional changes		
Command Signals: CI3–CI0 :		Am7968 pin # 15–12 respectively		
		Am79168 pin # 3–6 respectively		
		no functional changes		
Control Signals:	$\overline{\text{RESET}}$	Am7968 pin # 8	pin deleted	
	CLS	Am7968 pin # 10	pin deleted	
	ACK	Am7968 pin # 1	ACK	Am79168 pin # 10
	STRB	Am7968 pin #2	STRBI	Am79168 pin # 9
	DMS	Am7968 pin # 9	DMS	Am79168 pin # 17
		functional change 8-bit & 10-bit words only, 9-bit word deleted		
New features:				
	CDSEL	Am79168 pin # 7	Command or Data select input	
	$\overline{\text{OE1}}$	Am79168 pin # 11	output enable for TX/TY serial lines	
	$\overline{\text{OE2}}$	Am79168 pin # 12	output enable for TLOOP serial lines	
	$\overline{\text{ESEL}}$	Am79168 pin # 19	encoder select input	
Miscellaneous:	SERIN	Am7968 pin # 11	pin deleted	
	X2	Am7968 pin # 19	pin deleted	
	CLK	Am7968 pin # 18	pin deleted see description below	
	TEST	Am79168 pin # 13	factory test pin	
Power Supply:	V_{CC3}, V_{CC2}, V_{CC1}	Am7968 pin # 7, 5, 6	have been reduced to 2 pins	
	PV_{CC}, CV_{CC}	Am79168 pin # 15, 8		
	GND2, GND1	Am7968 pin # 21, 22	PGND, CGND	Am79168 pin # 21, 22

The Am79168 has several pin definition differences that enhance the operational ease of the TAXIchip Transmitter interface. The $\overline{\text{RESET}}$ function is no longer needed due to the robustness of the TAXI PLL structure. The cascading function that involves the CLS and SERIN pins is no longer on-chip because more cost effective solutions can be implemented with external logic (see Chapter 6 for suggestions). The CLK and X2 functions are no longer necessary because the crystal/TTL clock source options have been replaced with a TTL level only clock source. The buffered clock source for the CLKI input can replace the CLK output in designs that require external timing that is synchronous to the Transmitter.

A more user friendly Command generation structure has been implemented in the Am79168. The CDSEL input controls whether a Command or Data is to be sent. The Am7968 family requires that all Command lines be set to 0 before a data byte can be strobed into the TAXI device to be transmitted. The simple switching of one pin, CDSEL, before strobing is all that is required with the Am79168. The Am79168 can transmit synchronization characters (K28.5) as Commands or by no-strobing as with the Am7968 devices.

The serial interface and encoding logic have some very useful enhancements. The serial outputs on the Am79168 can be controlled to allow a "wired-OR" with the use of the $\overline{\text{OE1}}$ and $\overline{\text{OE2}}$ inputs. A diagnostic feature that has been added to the Am79168 is the TLOOP single-ended serial line. The TLOOP serial output can be connected to the Receiver RLOOP serial input to allow the testing of a TAXI pair in a network system node without the use of external link components. The T/RLOOP feature can also be used as an additional link, if the feature is not used for node diagnostics in a network system. The Am79168 also has a feature that allows the byte-by-byte disabling of the encoder with the $\overline{\text{ESEL}}$ input. Using the $\overline{\text{ESEL}}$ input, a single invalid symbol or a series of invalid symbols can be transmitted over a network system. The robustness of a network system can be evaluated using this feature. The $\overline{\text{ESEL}}$ input also makes it possible for the customer to use an external coding scheme.

Table 7.2 Receiver—Am7969 vs. Am79169-75

Differences in Pinouts and Related Functionality (Pinouts for PLCC Packages)

Pin #	Am7969	Am79169	Comments
1	DO3	DO6	
2	DO2	PGND	
3	DO1	DO5	
4	DO0	DO4	
5	IGM	DO3	IGM not needed, no on-chip cascading with Am79169
6	$\overline{\text{RESET}}$	DO2	$\overline{\text{RESET}}$ no longer needed
7	VCC1	DO1	
8	VCC2	DO0	
9	SERIN+	CGND	
10	SERIN-	CLKO	CLKO same as Am7969 CLK
11	DMS	REFCK	REFCK a TTL clock reference *similar* to X1 on Am7969
12	DSTRB	RLOOP	DSTRB, CSTRB & VLTN replaced with S1, S2 and STRBO functional difference, refer to Am79168/9-275 Data Sheet
13	CSTRB	RAW	RAW input provides option to disable decoder
14	VLTN	RY	
15	CO0	$\overline{\text{LPSEL}}$	$\overline{\text{LPSEL}}$ selects RX/RY or RLOOP input as decoded serial data stream
16	CO1	RX	
17	CO2/DO9	DMS	DMS with 8-bit and 10-bit word selection
18	CO3/DO8	SYNCD	SYNCD allows byte resynchronization to be disabled
19	CLK	STROBO	
20	GND1	S2	
21	GND2	CVCC	
22	X1	S1	X1 no longer necessary
23	X2	CO0	
24	CNB	CO1	CNB not needed, no on-chip cascading with Am79169
25	DO7	CO2/DO9	
26	DO6	CO3/DO8	
27	DO5	DO7	
28	DO4	PVCC	

Signal Group Differences

Data Signals: DO7–DO0:		Am7969 pin # 25–28, 1–4 respectively	
		Am79169 pin # 27,1, 3–8 respectively	
		no functional changes	
Command Signals: CO3–CO0:		Am7969 pin # 18–15 respectively	
		Am79169 pin # 26–23 respectively	
		no functional changes	
Control Signals:	IGM, CNB	Am7969 pin # 5,24	pins deleted
	$\overline{\text{RESET}}$	Am7969 pin # 6	pin deleted
	DMS	Am7969 pin # 11	DMS Am79169 pin # 17
		functional change 8-bit & 10-bit words only, 9-bit word deleted	
	DSTRB, CSTRB, VLTN	Am7969 pin # 12,13,14 functional change to S1, S2, STRBO Am79169 pin # 22,20,19	
New features:			
	SYNCD	Am79169 pin # 18	byte resynchronization disable input
	$\overline{\text{LPSEL}}$	Am79169 pin # 15	RLOOP serial line data stream select input
	RAW	Am79169 pin # 13	decoder disable input
Miscellaneous:	X2	Am7969 pin # 23	pin deleted
	CLK	Am7969 pin # 19	CLKO Am79169 pin # 10
Power Supply:	V_{CC2}, V_{CC1}	Am7969 pin # 8, 7	PV_{CC}, CV_{CC} Am79169 pin # 28, 21
	GND2, GND1	Am7969 pin # 21, 20	PGND, CGND Am79169 pin # 2, 9

The Am79169 has some pin definition differences that enhance the operation of the TAXIchip Receiver interface. The cascade function that involves the IGM and CNB pins is no longer on-chip because more cost effective solutions can be implemented with external logic (see Chapter 6 for suggestions). The $\overline{\text{RESET}}$ function is no longer needed due to the robustness of the TAXI PLL structure. The X2 input is no longer necessary because the crystal/TTL clock source options have been replaced with a TTL only clock source.

A diagnostic feature on the Am79169 that is not available on the Am7969 is the RLOOP single-ended serial line. This serial line input can be connected to the Transmitter TLOOP serial output to allow the testing of a TAXI pair used in a network system node without the use of external link components. The T/RLOOP feature can also be used as an additional link, if the feature is not used for node diagnostics in a network system.

The Am79169 also has a feature that allows the hardware disabling of the decoder with the RAW input. The RAW input allows the customer to use an external coding scheme if desired.

The byte resynchronization of the TAXI receiver has been enhanced with the Am79169. The realignment of the byte boundary for the Am79169 requires that two K28.5 symbols be received. The two K28.5 symbols can be separated by zero, one, two or three bytes of Data/Commands. The Am79169 is less likely to decode an alias Resync using this method of byte realignment than the previous TAXI devices.

For details on the operation of the above mentioned features please refer to the Am79168/9-275 Data Sheet PID# 15765A.

APPENDIX

A TEST MODE OPERATION

A.1 INTRODUCTION

Test mode operation exists on the TAXI-275 devices for factory testing of the internal logic. In Test Mode, the PLLs of the Transmitter and the Receiver are by-passed and the internal baud rate clock is applied from an external source. This allows the TAXI-275 to function at much slower speeds. This mode was designed to simplify the testing of TAXIchips in an automatic tester (ATE) production environment. When using Test Mode there is no minimum frequency.

Since the multiplying PLL is turned off in Test Mode, an external clock source must be supplied to the TAXIchips. In normal (non-test) mode, the Transmitter PLL multiplies the byte clock by 10 or 12. The new 10X clock or 12X clock is called the baud rate clock or baudclk, and is used to transmit the serial data. The Receiver PLL generates the same type of baudclk to decode the incoming data and to track and follow any fluctuations in the transmission frequency of the incoming data.

In test mode, the Transmitter PLL is disconnected and the internal clock multiplier is switched out. The internal logic is now clocked directly by the signal applied to the CLKI pin. The input to the CLKI pin now becomes the baudclk and must be supplied by the user. This baudclk must be 10X the byte rate for byte mode and 12X the byte rate for 10-bit mode.

On the Receiver side, the internal data tracking PLL is disconnected in Test Mode. An external clock recovery PLL circuit must be used to allow the Receiver to track the incoming serial data stream. This recovered baudclk is supplied to the REFCK input. Either a digital PLL or an analog PLL can be used for clock recovery as shown in Figure 7.2 Receiver Test Mode Connections.

A.2 TRANSMITTER CONNECTIONS

The CLKI pin is now an input for the baudclk (the baud rate clock). This means that if the serial transmission rate is to be 1.5 Kbaud, CLKI must be 1.5 KHz.

The ACK output is raised only when a byte has been successfully captured in the Transmitter input register and is ready for transmission. If STRBI is lowered before ACK is seen, ACK will be suppressed, see the STRBI/ACK description in Section 3.0).

The TEST input is also used as the reset pin for the internal state machines. For testing purposes, the following steps are to be taken upon power-up or initialization.

- The TEST input should be brought High and the Transmitter baudclk (CLKI) pulsed at least once.
- The TEST input should be brought to $V_{CC}/2$ or floating and the Transmitter baudclk pulsed at least once to reset the internal state machines.
- The TEST input should be set to the High state for the duration of test.

This serves to flush all extraneous data from the buffers and reset all internal state machines and leave the TAXI-275 in Test Mode. Once this is completed the Transmitter may be Strobed. The serial lines will then transmit data at the rate of the CLKI frequency.

The STRBI input must now be strobed only once every 10 baudclk (CLKI) pulses or once every 12 baudclk pulses depending upon the mode selected. This will allow time for an 8-bit word to be encoded to 10 bits or a 10-bit word to be encoded to 12 bits and shifted out one bit every clock pulse.

The parallel data input pins are provided with new data every 10/12 baudclk pulses. Setup and hold times remain the same as in non-Test Mode with respect to STRBI. In the non-Test modes, the clock rate is the byte rate and a new data word and a strobe is provided every clock pulse. In test mode, the clock rate is the baud rate so the new data word and strobe are provided every n clock pulses.

In Test Mode, the Transmitter outputs encoded data on the TX, TY and TLOOP serial pins. All of the serial outputs must be terminated properly to electrically balance the outputs.

When the DMS input is hardwired to $V_{CC}/2$ or left floating, the Receiver will be in Test Mode. Test Mode on the Receiver only operates in 8-bit mode.

The REFCK input is then the baud clock input (baudclk), just like the CLKI input on the Transmitter.

The CLKO output remains a byte rate clock out.

A.3 TIMING RELATIONSHIPS IN TEST MODE

The timing parameters in Test Mode are similar to the parameters in standard mode. Propagation delay values remain the same, however baud interval relationships are now calculated with respect to the new baud rates. Please refer to Am79168/9-275 Data Sheet for details.

APPENDIX

B APPLICATION DESIGN NOTES

B.1 INTRODUCTION TO THE EMERGING HIGH SPEED DATA COMMUNICATION STANDARDS

Communication between most large computing systems has been accomplished over short distances (usually within the space of one room) using parallel bus technologies or "channels". Large computing resources have an increasing need to communicate at high speed over long data links.

Recent developments in communication technologies permit very high speed digital transmission using both fiber and copper media over extended distances, at increasing level of reliability of the connection. In order to maximize the benefits of the bandwidth made available by the new technologies, new high speed serial communication protocols are being developed. These protocols are point-to-point in nature, resulting in switched networks with management capabilities. The attributes of the new communication protocols are not available with either traditional Local Area Network (LAN) technologies or parallel computer channels where the bandwidth is shared.

IBM™ is the first mainframe computer manufacturer to endorse the advancement of new technologies with Enterprise System Connection Architecture: (ESCON)™. ESCON is a switched fiber optical data channel. This represents a dramatic move in I/O interconnect architecture from the fundamentally unchanged channel architecture of the past. The ANSI X3T9.3 Fiber Channel Standard targets channel communication as well as offering high speed serial data transport for HIPPI (High Performance Parallel Interface), IPI (Intelligent Peripheral Interface), SCSI (Small Computer System Interface), and IEEE 802.2 LLC (Logical Link Control for local area networks).

As with any new technology, the initial deployment costs are high. As the market develops, the availability of technologies and competition will drive the cost downward. Eventually, many other applications, such as high performance work stations, will then be connected into these very high speed switched data networks, thus expanding the market even further.

Appendix E highlights some of the main features of ESCON and Fibre Channel and illustrates briefly how the two protocol systems are different. Where the AMD TAXI-275 8B/10B TAXI can fit into these applications as part of a cost effective solution is described here.

B.2 THE TAXI-275 AND ESCON/FIBRE CHANNEL

The typical ESCON node environment has an ESCON optical module, serializing/deserializing logic, encoder/decoder logic, a state machine that interprets/generates the input and output with ESCON protocol and an interface to the main system. The TAXI-275 Transmitter contains the required 8B10B encoder and the parallel-to-serial converter, while the receiver contains the required serial-to-parallel converter and 8B10B decoder. All of the high-speed clock (baud rate) generation and recovery are done on-chip with phase-locked loop (PLL) circuitry.

Figure B.1 A Typical ESCON Node Using TAXIchip

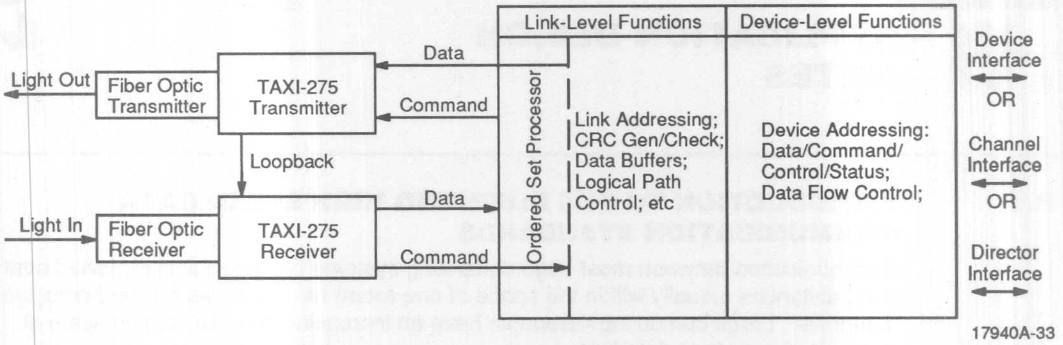
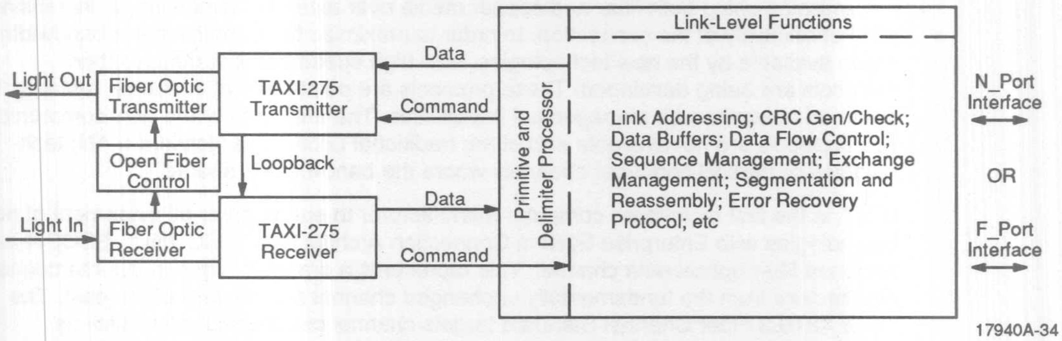


Figure B.2 A Typical Fibre Channel Node Using TAXIchip



The TAXI-275 chipset provides the higher level functions with a friendly and clean interface. The chipset provides physical layer data transfer functions of the ESCON Link-Level protocol (see Figure B.1) and a significant portion of the FC-1 of the Fibre Channel Standard (see Figure B.2). All ESCON and FCS special characters (active or reserved) are implemented as Commands and the TAXI automatically transmits ESCON Idle characters (K28.5) when there is no pending data transmission. All Fibre Channel delimiters and primitives can be generated using TAXI devices with one simple interface.

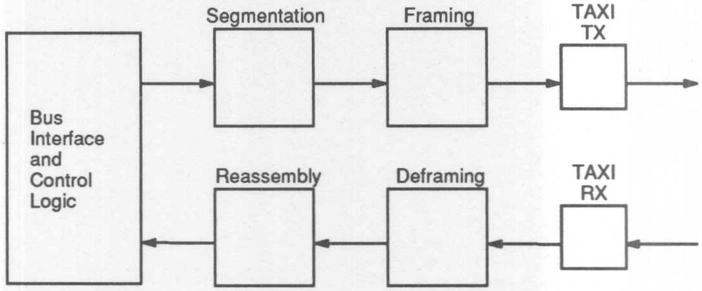
Currently, the TAXI-275 devices are specified to operate from 175 to 275 MBaud. This range covers the ESCON rate of 200 MBaud and the quarter-speed FCS rate of 265.625 MBaud.

B.3 ASYNCHRONOUS TRANSFER MODE (ATM)

ATM has the potential for a broad range of applications from Local Area Network (LAN) connections to the originally intended market of Wide Area Network (WAN) connections. The market also includes private and public networks. Two of the network standards that ATM can work with are SONNET and B-ISDN (still under standardization). The future potential of ATM is to allow connections between mainframes, workstations, and personal computers with a simple phone call.

The TAXI-275 chipset can be used in ATM connections. The ATM physical layer supports 4B5B and 8B10B coding schemes in its hubs, bridges, routers, switches and terminals with some restrictions. The TAXI-275 fits into the physical layer as the coding and serializing chipset for the 155.52 Mbit/s frame rate (194 MBaud serial line rate). Figure B.3 illustrates the functional role of the TAXI-275 devices.

Figure B.3 Block Diagram of a TAXI-275 ATM Node



17940A-35

APPENDIX

C VENDORS AND MANUFACTURERS LISTING

C.1 FIBER OPTIC MEDIA

Presented below is a partial listing of fiber optic data link suppliers that manufacture or market optical components in a data rate range compatible with the TAXI-275. Several of these components have been demonstrated in a bench level evaluation in conjunction with the TAXIchip devices.

AMP
Pennsylvania
(717) 986-5106 Literature
(908) 685-2000 Technical, Litel Division

AT&T Technologies Literature and referral to
1-800-372-2447 optical modules and fiber media

BT&D Technologies
Delaware
(800) 545-4306

Sumitomo Electric
New York
(914) 347-3770

Hewlett Packard Literature and
800 752-0900 referral to optical components

NEC Electronics, Inc.
1-800-632-3531 Literature
(415) 960-6000 Technical

Plessey
United Kingdom
011-44-327-51871

Siemens
New Jersey
1-800-827-3334

C.2 COPPER MEDIA

These are just a couple of examples of companies that supply copper media.

Manu.	= Belden Electronics
Dist.	= determined by location
Phone #	= 1-800-235-3361
State	= local to customer
Example	= STP Type 1, part # 9688

Manu.	= Anixter
Dist.	= determined by phone
Phone #	= (800) 323-8166
State	= local to customer
Example	= none

C.3 CRYSTAL OSCILLATORS

The following list contains the crystal oscillator manufactures that have been used with TAXI devices in the evaluation process of the TAXI-275 parts. The specifications require a crystal based oscillator with no more than a 60/40 duty cycle, optimally a 50/50 duty cycle, and a frequency tollerance of ±0.1%. Frequencies that are nonstandard (other than 20 MHz or 25 MHz) will require special manufacturing. There is usually an approximate turn-around-time of 4 to 6 weeks when special ordering these oscillators. This time must be accounted for in the design schedule. Some product examples are included for your reference.

Manu.	= RXD
Dist.	= determined by location
Phone #	= (800) 228-8108
State	= local to customer
Example	= 20 MHz crystal Oscillator, part #: RXDT-2

Manu.	= M.F. Electronics
Dist.	= M.F. Electronics
Phone #	= (914) 576-6570
State	= New York
Example	= crystal oscillator, part # : M1200 26.5625 MHz

C.4 CAPACITORS, FERRITE BEADS, and TRANSFORMERS

Capacitors

The capacitors for use with the TAXI-275 have similar specifications as the previous TAXI family members. The goal in selecting an AC coupling capacitor is for the capacitor to have as low an inductive (L) and resistive (R) value as possible at high frequencies. The lower the L and R values the clearer the serial signal after passing through the capacitor. Use high quality RF grade capacitors. Type COG, NPO or x7R have the recommended characteristics. The use of Z5U capacitors is not recommended.

Ferrite Beads

The ferrite bead list below consists of the units that are used in previous designs using the TAXI-275 devices. The impedence value at a specific frequency is chosen based on the expected characteristics of the power supply. The ferrite bead filters noise of a specific frequency range; experimentation may be necessary under different circumstances.

Manu.	=	Fair-Rite
Dist.	=	Power Supply Components
Phone #	=	(408)737-1333
State	=	California
Example	=	part # 2743002111
		part # 2743021447 SM Ferrite Bead

Transformers

The transformers that are used with the TAXI-275 device are specially manufactured by the company below for compatibility with Fibre Channel specifications. At this time Pulse Engineering is the only supplier of these devices.

Manu.	=	Pulse Engineering Inc.
Contact	=	Mukesh Mehta
Phone #	=	(619) 674-8194
State	=	San Diego, CA
Example	=	PE 65507 for $Z_o = 75\ \Omega$
		PE 65506 for $Z_o = 150\ \Omega$

Manu.	=	Technitrol
Phone #	=	(215) 426-9105
State	=	Philadelphia, PA
Example	=	

APPENDIX

D DATA ENCODING

D.1 8B10B TRANSMISSION BLOCK CODE

D.1.1 Introduction

The Fibre Channel 8B10B code was invented and patented by IBM[1,2] in 1984. The details can be found in US patents and IBM publications[3]. This data transmission code had been adopted by ANSI X3T9.3 Fibre Channel working group in the FC_PH document and will become part of the standard. The same transmission code is also used in ESCON (Enterprise System Connection) by IBM.

The basic requirements for a self-clocking data transmission code are limited run-length, high transition density, fixed block size, and DC balance. The 8B10B code is a block code which encodes 8-bit user data blocks into 10-bit code words for serial transmission. It satisfies many of the requirements—freedom of the DC component is its most unique feature. Maintaining DC balance for data transmission over optical fiber or copper media is of particular importance at high data rates.

D.1.2 Disparity and Running Disparity

The measurement of the DC component in any serially transmitted bit stream is the number of ones (high voltage level) and number of zeros (low voltage level) that are transmitted over the medium. Disparity is defined as the difference between the numbers of ones and zeros in a fixed size transmission block. For example, for a block of 10 bits,

0101011001	5 ones and 5 zeros	→	disparity = 0
0111001011	6 ones and 4 zeros	→	disparity = +2
0100010110	4 ones and 6 zeros	→	disparity = -2

In order to maintain transmission without the DC component, it is an obvious goal to keep the disparity of each data block to the minimum. The 8B10B code does not use any code word with disparity other than +2, 0, or -2.

Running Disparity (RD) is the running (cumulative) sum of the disparity of all previously transmitted data blocks. RD is a measurement of the amount of DC component in the transmitted signal. For example, assuming an artificially determined initial RD of -1,

Initial	0101011001	0111001011	0100010110
RD	disparity = 0	disparity = +2	disparity = -2
(-1)	(-1)	(+1)	(-1)

the current RD values after each transmitted block is shown within each pair of ( ).

By limiting the allowable RD during data transmission, one is able to control the DC imbalance of a transmission code. The Running Disparity rule of the 8B10B code mandates the choice of code words such that:

- a code word with disparity=0 can be sent regardless of current RD,
- a code word with disparity=+2 can be sent only if current RD is -1,
- a code word with disparity=-2 can be sent only if current RD is +1.

Assuming initial RD = -1, the only possible values of RD at any code block boundary are +1 and -1, as shown by the earlier example.

D.1.3 Principles

The 8B10B encode/decode is done in two subblocks, 5B6B and 3B4B, concatenated together. A block of 8-bit user data (byte) is divided into a 5-bit subblock and a 3-bit subblock. The 5-bit subblock is encoded into a 6-bit NRZ code word and the 3-bit subblock is encoded into a 4-bit NRZ code word. Each encoded subblock code word can have an equal number of ones and zeros (disparity=0), more ones than zeros (disparity = +2), or more zeros than ones (disparity = -2). The 8B10B code does not use subblock code words with other disparity values such as ±4 or ±6.

The Running Disparity rule requires that the 5B6B and 3B4B codes encode user data in such a way that

- a code word with disparity = 0 is used to uniquely represent a byte of data, or
- a code word with disparity = +2 and its binary complement (with disparity = -2) are used together to represent one byte of data → a pair of complementary symbols represent the same byte of data.

In the 5B6B subblock (see Table D.1), the data space is $2^5 = 32$ and the code space is $2^6 = 64$. This means that 32 out of the 64 potential 6-bit code words are needed to represent the data. The break down of the 64 code words is as follows: 20 each of the possible code words are zero-disparity (3 ones and 3 zeros), 15 pairs are ±2 disparity (4 ones and 2 zeros or 2 ones and 4 zeros), and 14 violate the run length limit of the code. Eighteen of the zero-disparity code words are used to represent eighteen of the 5-bit user data bytes; the remaining two zero-disparity code words are used as a pair to represent one 5-bit user data byte. Thirteen of the ±2 disparity code pairs are used to represent 13 of the 5-bit user data bytes; one of the 15 pairs is used in the special character K28 which will be discussed later, one pair is not used.
(18 + 1 + 13 + 1(K28) = 33)

In the 3B4B subblock (see Table D.1), the data space is $2^3 = 8$ and the code space is $2^4 = 16$. This means that 8 out of the 16 potential 4-bit code words are needed to represent the data. The break down of the 16 code words is as follows: 6 each of the possible code words are zero-disparity (2 ones and 2 zeros), 4 pairs are ±2 disparity (3 ones and 1 zero or 1 one and 3 zeros), and 2 violate the run length limit of the code. Four of the zero-disparity code words are used to represent four of the 4-bit user data bytes; the remaining two zero-disparity code words are used as a pair to represent one 4-bit user data byte. Three of the ±2 disparity code pairs are used to represent three of the 4-bit user data bytes (includes exception for 111), one is not used. (4 + 1 + 3 = 8).

The individual assignments of the code words to each bit pattern is theoretically arbitrary. In practice however, it is done in such a way that the combinatorial logic of the encoder/decoder is minimized. For the complete 8B10B Encode/Decode table refer to Table D.2.

Table D.1 8B10B Encode/Decode Table

5B6B Subblock Table (left-most encoded bit is transmitted first serially)

5-Bit Data	Encoded	(Disparity)	Alternate	(Disparity)
00000	011000	(-2)	100111	(+2)
00001	100010	(-2)	011101	(+2)
00010	010010	(-2)	101101	(+2)
00011	110001	(0)	none	
00100	001010	(-2)	110101	(+2)
00101	101001	(0)	none	
00110	011001	(0)	none	
00111	111000	(0)	000111	(0)
01000	000110	(-2)	111001	(+2)
01001	100101	(0)	none	
01010	010101	(0)	none	
01011	110100	(0)	none	
01100	001101	(0)	none	
01101	101100	(0)	none	
01110	011100	(0)	none	
01111	101000	(-2)	010111	(+2)
10000	011011	(+2)	100100	(-2)
10001	100011	(0)	none	
10010	010011	(0)	none	
10011	110010	(0)	none	
10100	001011	(0)	none	
10101	101010	(0)	none	
10110	011010	(0)	none	
10111	111010	(+2)	000101	(-2)
11000	001100	(-2)	110011	(+2)
11001	100110	(0)	none	
11010	010110	(0)	none	
11011	110110	(+2)	001001	(-2)
11100	001110	(0)	none	
11101	101110	(+2)	010001	(-2)
11110	011110	(+2)	100001	(-2)
11111	101011	(+2)	010100	(-2)

Table D.1 **8B10B Encode/Decode Table (continued)**

3B4B Subblock Table (left-most encoded bit is transmitted first serially)

3-Bit Data	Encoded	(Disparity)	Alternate	(Disparity)
000	0100	(-2)	1011	(+2)
001	1001	(0)	none	
010	0101	(0)	none	
011	1100	(0)	0011	(0)
100	0010	(-2)	1101	(+2)
101	1010	(0)	none	
110	0110	(0)	none	
111	1110	(+2)	0001	(-2)
111(exception)	0111	(+2)	1000	(-2)

Notes:

1. *"Alternate" is always the complement of the "Encoded".*
2. *The "Encoded" without "Alternate" are all zero-disparity code words.*
3. *Running Disparity rule mandates the choice of a code word or its alternate.*

Table D.2. Complete 8B10B Encoding Table

Data Byte Name	Bits HGF	EDCBA	Current RD -		Current RD +	
			abcdei	fghj	abcdei	fghj
D0.0	000	00000	100111	0100	011000	1011
D1.0	000	00001	011101	0100	100010	1011
D2.0	000	00010	101101	0100	010010	1011
D3.0	000	00011	110001	1011	110001	0100
D4.0	00	00100	110101	0100	001010	1011
D5.0	000	00101	101001	1011	101001	0100
D6.0	000	00110	011001	1011	011001	0100
D7.0	000	00111	111000	1011	000111	0100
D8.0	000	01000	111001	0100	000110	1011
D9.0	000	01001	100101	1011	100101	0100
D10.0	000	01010	010101	1011	010101	0100
D11.0	000	01011	110100	1011	110100	0100
D12.0	000	01100	001101	1011	001101	0100
D13.0	000	01101	101100	1011	101100	0100
D14.0	000	01110	011100	1011	011100	0100
D15.0	000	01111	010111	0100	101000	1011
D16.0	000	10000	011011	0100	100100	1011
D17.0	000	10001	100011	1011	100011	0100
D18.0	000	10010	010011	1011	010011	0100
D19.0	000	10011	110010	1011	110010	0100
D20.0	000	10100	001011	1011	001011	0100
D21.0	000	10101	101010	1011	101010	0100
D22.0	000	10110	011010	1011	011010	0100
D23.0	000	10111	111010	0100	000101	1011
D24.0	000	11000	110011	0100	001100	1011
D25.0	000	11001	100110	1011	100110	0100
D26.0	000	11010	010110	1011	010110	0100
D27.0	000	11011	110110	0100	001001	1011
D28.0	000	11100	001110	1011	001110	0100
D29.0	000	11101	101110	0100	010001	1011
D30.0	000	11110	011110	0100	100001	1011
D31.0	000	11111	101011	0100	010100	1011

Table D.2. Complete 8B10B Encoding Table (continued)

Data Byte Name	Bits HGF	Bits EDCBA	Current RD -		Current RD +	
			abcdei	fghj	abcdei	fghj
D0.1	001	00000	100111	1001	011000	1001
D1.1	001	00001	011101	1001	100010	1001
D2.1	001	00010	101101	1001	010010	1001
D3.1	001	00011	110001	1001	110001	1001
D4.1	001	00100	110101	1001	001010	1001
D5.1	001	00101	101001	1001	101001	1001
D6.1	001	00110	011001	1001	011001	1001
D7.1	001	00111	111000	1001	000111	1001
D8.1	001	01000	111001	1001	000110	1001
D9.1	001	01001	100101	1001	100101	1001
D10.1	001	01010	010101	1001	010101	1001
D11.1	001	01011	110100	1001	110100	1001
D12.1	001	01100	001101	1001	001101	1001
D13.1	001	01101	101100	1001	101100	1001
D14.1	001	01110	011100	1001	011100	1001
D15.1	001	01111	010111	1001	101000	1001
D16.1	001	10000	011011	1001	100100	1001
D17.1	001	10001	100011	1001	100011	1001
D18.1	001	10010	010011	1001	010011	1001
D19.1	001	10011	110010	1001	110010	1001
D20.1	001	10100	001011	1001	001011	1001
D21.1	001	10101	101010	1001	101010	1001
D22.1	001	10110	011010	1001	011010	1001
D23.1	001	10111	111010	1001	000101	1001
D24.1	001	11000	110011	1001	001100	1001
D25.1	001	11001	100110	1001	100110	1001
D26.1	001	11010	010110	1001	010110	1001
D27.1	001	11011	110110	1001	001001	1001
D28.1	001	11100	001110	1001	001110	1001
D29.1	001	11101	101110	1001	010001	1001
D30.1	001	11110	011110	1001	100001	1001
D31.1	001	11111	101011	1001	010100	1001

Table D.2. Complete 8B10B Encoding Table (continued)

Data Byte Name	Bits HGF	Bits EDCBA	Current RD -		Current RD +	
			abcdei	fghj	abcdei	fghj
D0.2	010	00000	100111	0101	011000	0101
D1.2	010	00001	011101	0101	100010	0101
D2.2	010	00010	101101	0101	010010	0101
D3.2	010	00011	110001	0101	110001	0101
D4.2	010	00100	110101	0101	001010	0101
D5.2	010	00101	101001	0101	101001	0101
D6.2	010	00110	011001	0101	011001	0101
D7.2	010	00111	111000	0101	000111	0101
D8.2	010	01000	111001	0101	000110	0101
D9.2	010	01001	100101	0101	100101	0101
D10.2	010	01010	010101	0101	010101	0101
D11.2	010	01011	110100	0101	110100	0101
D12.2	010	01100	001101	0101	001101	0101
D13.2	010	01101	101100	0101	101100	0101
D14.2	010	01110	011100	0101	011100	0101
D15.2	010	01111	010111	0101	101000	0101
D16.2	010	10000	011011	0101	100100	0101
D17.2	010	10001	100011	0101	100011	0101
D18.2	010	10010	010011	0101	010011	0101
D19.2	010	10011	110010	0101	110010	0101
D20.2	010	10100	001011	0101	001011	0101
D21.2	010	10101	101010	0101	101010	0101
D22.2	010	10110	011010	0101	011010	0101
D23.2	010	10111	111010	0101	000101	0101
D24.2	010	11000	110011	0101	001100	0101
D25.2	010	11001	100110	0101	100110	0101
D26.2	010	11010	010110	0101	010110	0101
D27.2	010	11011	110110	0101	001001	0101
D28.2	010	11100	001110	0101	001110	0101
D29.2	010	11101	101110	0101	010001	0101
D30.2	010	11110	011110	0101	100001	0101
D31.2	010	11111	101011	0101	010100	0101

Table D.2. **Complete 8B10B Encoding Table (continued)**

Data Byte Name	Bits HGF	Bits EDCBA	Current RD -		Current RD +	
			abcdei	fghj	abcdei	fghj
D0.3	011	00000	100111	0011	011000	1100
D1.3	011	00001	011101	0011	100010	1100
D2.3	011	00010	101101	0011	010010	1100
D3.3	011	00011	110001	1100	110001	0011
D4.3	011	00100	110101	0011	001010	1100
D5.3	011	00101	101001	1100	101001	0011
D6.3	011	00110	011001	1100	011001	0011
D7.3	011	00111	111000	1100	000111	0011
D8.3	011	01000	111001	0011	000110	1100
D9.3	011	01001	100101	1100	100101	0011
D10.3	011	01010	010101	1100	010101	0011
D11.3	011	01011	110100	1100	110100	0011
D12.3	011	01100	001101	1100	001101	0011
D13.3	011	01101	101100	1100	101100	0011
D14.3	011	01110	011100	1100	011100	0011
D15.3	011	01111	010111	0011	101000	1100
D16.3	011	10000	011011	0011	100100	1100
D17.3	011	10001	100011	1100	100011	0011
D18.3	011	10010	010011	1100	010011	0011
D19.3	011	10011	110010	1100	110010	0011
D20.3	011	10100	001011	1100	001011	0011
D21.3	011	10101	101010	1100	101010	0011
D22.3	011	10110	011010	1100	011010	0011
D23.3	011	10111	111010	0011	000101	1100
D24.3	011	11000	110011	0011	001100	1100
D25.3	011	11001	100110	1100	100110	0011
D26.3	011	11010	010110	1100	010110	0011
D27.3	011	11011	110110	0011	001001	1100
D28.3	011	11100	001110	1100	001110	0011
D29.3	011	11101	101110	0011	010001	1100
D30.3	011	11110	011110	0011	100001	1100
D31.3	011	11111	101011	0011	010100	1100

Table D.2. Complete 8B10B Encoding Table (continued)

Data Byte Name	Bits HGF	Bits EDCBA	Current RD -		Current RD +	
			abcdei	fghj	abcdei	fghj
D0.4	100	00000	100111	0010	011000	1101
D1.4	100	00001	011101	0010	100010	1101
D2.4	100	00010	101101	0010	010010	1101
D3.4	100	00011	110001	1101	110001	0010
D4.4	100	00100	110101	0010	001010	1101
D5.4	100	00101	101001	1101	101001	0010
D6.4	100	00110	011001	1101	011001	0010
D7.4	100	00111	111000	1101	000111	0010
D8.4	100	01000	111001	0010	000110	1101
D9.4	100	01001	100101	1101	100101	0010
D10.4	100	01010	010101	1101	010101	0010
D11.4	100	01011	110100	1101	110100	0010
D12.4	100	01100	001101	1101	001101	0010
D13.4	100	01101	101100	1101	101100	0010
D14.4	100	01110	011100	1101	011100	0010
D15.4	100	01111	010111	0010	101000	1101
D16.4	100	10000	011011	0010	100100	1101
D17.4	100	10001	100011	1101	100011	0010
D18.4	100	10010	010011	1101	010011	0010
D19.4	100	10011	110010	1101	110010	0010
D20.4	100	10100	001011	1101	001011	0010
D21.4	100	10101	101010	1101	101010	0010
D22.4	100	10110	011010	1101	011010	0010
D23.4	100	10111	111010	0010	000101	1101
D24.4	100	11000	110011	0010	001100	1101
D25.4	100	11001	100110	1101	100110	0010
D26.4	100	11010	010110	1101	010110	0010
D27.4	100	11011	110110	0010	001001	1101
D28.4	100	11100	001110	1101	001110	0010
D29.4	100	11101	101110	0010	010001	1101
D30.4	100	11110	011110	0010	100001	1101
D31.4	100	11111	101011	0010	010100	1101

Table D.2. Complete 8B10B Encoding Table (continued)

Data Byte Name	Bits HGF	Bits EDCBA	Current RD -		Current RD +	
			abcdei	fghj	abcdei	fghj
D0.5	101	00000	100111	1010	011000	1010
D1.5	101	00001	011101	1010	100010	1010
D2.5	101	00010	101101	1010	010010	1010
D3.5	101	00011	110001	1010	110001	1010
D4.5	101	00100	110101	1010	001010	1010
D5.5	101	00101	101001	1010	101001	1010
D6.5	101	00110	011001	1010	011001	1010
D7.5	101	00111	111000	1010	000111	1010
D8.5	101	01000	111001	1010	000110	1010
D9.5	101	01001	100101	1010	100101	1010
D10.5	101	01010	010101	1010	010101	1010
D11.5	101	01011	110100	1010	110100	1010
D12.5	101	01100	001101	1010	001101	1010
D13.5	101	01101	101100	1010	101100	1010
D14.5	101	01110	011100	1010	011100	1010
D15.5	101	01111	010111	1010	101000	1010
D16.5	101	10000	011011	1010	100100	1010
D17.5	101	10001	100011	1010	100011	1010
D18.5	101	10010	010011	1010	010011	1010
D19.5	101	10011	110010	1010	110010	1010
D20.5	101	10100	001011	1010	001011	1010
D21.5	101	10101	101010	1010	101010	1010
D22.5	101	10110	011010	1010	011010	1010
D23.5	101	10111	111010	1010	000101	1010
D24.5	101	11000	110011	1010	001100	1010
D25.5	101	11001	100110	1010	100110	1010
D26.5	101	11010	010110	1010	010110	1010
D27.5	101	11011	110110	1010	001001	1010
D28.5	101	11100	001110	1010	001110	1010
D29.5	101	11101	101110	1010	010001	1010
D30.5	101	11110	011110	1010	100001	1010
D31.5	101	11111	101011	1010	010100	1010

Table D.2. Complete 8B10B Encoding Table (continued)

Data Byte Name	Bits HGF	Bits EDCBA	Current RD -		Current RD +	
			abcdei	fghj	abcdei	fghj
D0.6	110	00000	100111	0110	011000	0110
D1.6	110	00001	011101	0110	100010	0110
D2.6	110	00010	101101	0110	010010	0110
D3.6	110	00011	110001	0110	110001	0110
D4.6	110	00100	110101	0110	001010	0110
D5.6	110	00101	101001	0110	101001	0110
D6.6	110	00110	011001	0110	011001	0110
D7.6	110	00111	111000	0110	000111	0110
D8.6	110	01000	111001	0110	000110	0110
D9.6	110	01001	100101	0110	100101	0110
D10.6	110	01010	010101	0110	010101	0110
D11.6	110	01011	110100	0110	110100	0110
D12.6	110	01100	001101	0110	001101	0110
D13.6	110	01101	101100	0110	101100	0110
D14.6	110	01110	011100	0110	011100	0110
D15.6	110	01111	010111	0110	101000	0110
D16.6	110	10000	011011	0110	100100	0110
D17.6	110	10001	100011	0110	100011	0110
D18.6	110	10010	010011	0110	010011	0110
D19.6	110	10011	110010	0110	110010	0110
D20.6	110	10100	001011	0110	001011	0110
D21.6	110	10101	101010	0110	101010	0110
D22.6	110	10110	011010	0110	011010	0110
D23.6	110	10111	111010	0110	000101	0110
D24.6	110	11000	110011	0110	001100	0110
D25.6	110	11001	100110	0110	100110	0110
D26.6	110	11010	010110	0110	010110	0110
D27.6	110	11011	110110	0110	001001	0110
D28.6	110	11100	001110	0110	001110	0110
D29.6	110	11101	101110	0110	010001	0110
D30.6	110	11110	011110	0110	100001	0110
D31.6	110	11111	101011	0110	010100	0110

Table D.2. Complete 8B10B Encoding Table (continued)

Data Byte Name	Bits HGF	Bits EDCBA	Current RD -		Current RD +	
			abcdei	fghj	abcdei	fghj
D0.7	111	00000	100111	0001	011000	1110
D1.7	111	00001	011101	0001	100010	1110
D2.7	111	00010	101101	0001	010010	1110
D3.7	111	00011	110001	1110	110001	0001
D4.7	111	00100	110101	0001	001010	1110
D5.7	111	00101	101001	1110	101001	0001
D6.7	111	00110	011001	1110	011001	0001
D7.7	111	00111	111000	1110	000111	0001
D8.7	111	01000	111001	0001	000110	1110
D9.7	111	01001	100101	1110	100101	0001
D10.7	111	01010	010101	1110	010101	0001
D11.7	111	01011	110100	1110	110100	1000
D12.7	111	01100	001101	1110	001101	0001
D13.7	111	01101	101100	1110	101100	1000
D14.7	111	01110	011100	1110	011100	1000
D15.7	111	01111	010111	0001	101000	1110
D16.7	111	10000	011011	0001	100100	1110
D17.7	111	10001	100011	0111	100011	0001
D18.7	111	10010	010011	0111	010011	0001
D19.7	111	10011	110010	1110	110010	0001
D20.7	111	10100	001011	0111	001011	0001
D21.7	111	10101	101010	1110	101010	0001
D22.7	111	10110	011010	1110	011010	0001
D23.7	111	10111	111010	0001	000101	1110
D24.7	111	11000	110011	0001	001100	1110
D25.7	111	11001	100110	1110	100110	0001
D26.7	111	11010	010110	1110	010110	0001
D27.7	111	11011	110110	0001	001001	1110
D28.7	111	11100	001110	1110	001110	0001
D29.7	111	11101	101110	0001	010001	1110
D30.7	111	11110	011110	0001	100001	1110
D31.7	111	11111	101011	0001	010100	1110

D.1.4 An Example of 8B10B Encoding

The easiest way to understand the 8B10B code is to follow an example. Let there be four bytes of user data to be encoded.

HEX user data:	72	10	FF	0A
bit definitions:	HGF EDCBA	HGF EDCBA	HGF EDCBA	HGF EDCBA
binary user data:	011 10010	000 10000	111 11111	000 01010
bit definitions:	abcde fgh	abcde fgh	abcde fgh	abcde fgh
8B 10B subblocks:	01001 110	00001 000	11111 111	01010 000
8B10B notation:	D18 . 3	D16 . 0	D31 . 7	D10 . 0
initial RD:	(-1)			
5B6B encode; (RD)	010011 (-1)			
3B4B encode; (RD)	1100 (-1)			
5B6B encode; (RD)		011011 (+1)		
3B4B encode; (RD)		0100 (-1)		
5B6B encode; (RD)			101011 (+1)	
3B4B encode; (RD)			0001 (-1)	
5B6B encode; (RD)				010101 (-1)
3B4B encode; (RD)				1011 (+1)
final RD:	(+1)			

serial bit stream: 0100111100011011010010101100010101011011

Observations: within the 40-bit stream, there are 21 ones and 19 zeros. The value of Running Disparity changes from an initial -1 to a final +1, with intermediate values at subblock boundary of either -1 or +1. For each subblock, the encoding decision is made based upon the Running Disparity rule.

D.1.5 Special Characters

When the 5B6B subblock and 3B4B subblock are concatenated according to the RD rule, the resultant 10-bit code word is mapped into the $2^{10} = 1024$ code space. There are some code words which satisfy the run-length and transition density requirements although the code words are not used by legally encoded data. These code words (special characters) are available for representation of other data transmission functions, such as block alignment. They are:

Special Code Name	Current RD–		Current RD+		Notes
	abcdei	fghj	abcdei	fghhj	
K28.0	001111	0100	110000	1011	Reserved
K28.1	001111	1001	110000	0110	
K28.2	001111	0101	110000	1010	
K28.3	001111	0011	110000	1100	Reserved
K28.4	001111	0010	110000	1101	
K28.5	001111	1010	110000	0101	
K28.6	001111	0010	110000	1001	
K28.7	001111	1000	110000	0111	
K23.7	111010	1000	000101	0111	Reserved
K27.7	110110	1000	001001	0111	Reserved
K29.7	101110	1000	010001	0111	Reserved
K30.7	011110	1000	100001	0111	Reserved

All special character transmissions must follow the same Running Disparity rule. The special characters K28.1 and K28.5 have 5 consecutive ones following 2 zeros or 5 consecutive zeros following 2 ones. These combinations do not occur in a normally encoded data stream. These two special characters can be used for byte synchronization, i.e., the unique bit pattern is used to identify the byte boundary for serial-to-parallel conversion.

The Fibre Channel Standard has chosen K28.5 as part of all delimiters and primitives. ESCON employs more special characters than Fibre Channel to form "ordered sets" for the purpose of low-level signaling without encoded data transmission which involves upper layer protocols.

D.1.6 Exceptions to Encode/Decode Rules

There are a few exceptions to the simple data Encode/Decode rules in order to avoid the possibility of producing an alias special character from normally encoded data. The special characteristic of the "code alignment" character (K28.5) is the 5 consecutive ones or zeros which must be guaranteed never to occur in normally encoded and concatenated data characters.

The 5B6B code words that might cause an alias K28.5 are 000111 and 111000. The solution is to limit the usage of these code words by allowing only paired representation of data 00111. The same limitation applies to the 3B4B code word 0011 and 1100. The net effect for this limited usage of a zero-disparity code amounts to an association of somewhat artificial disparity value to these codes. Codes 000111 and 0011 are only legal when RD is +1; 111000 and 1100 are only legal when RD is -1.

The 3B4B code words that might cause an alias K28.5 are the code words 1110 and 0001. Therefore, these code words cannot be used freely as well. For a few cases listed below, the 3B4B codes used to represent data 111 must be changed to a different pair of code words, 0111 and 1000. The exceptions are:

5B6B 3B4B	Encoded	Alternate
01011 111	no exception	110100 1000
01101 111	no exception	101100 1000
01110 111	no exception	011100 1000
10001 111	100011 0111	no exception
10010 111	010011 0111	no exception
10100 111	001011 0111	no exception

D.1.7 Error Detection

8B10B code uses 452 (256 x 2–18 x 4 + 12) unique 10-bit code words to represent 256 8-bit data and 12 special characters. The number of codes used is derived from 2^8 user data patterns encoded into paired 10-bit primary and alternate code words with 18 zero-disparity 6-bit code words in combination with four zero-disparity 4-bit code words that do not have paired representations.

Any transmission error which transforms a valid code into one of the 572 (1024–452) invalid 10-bit code words will be detected by the combinatorial decode logic and flagged as an invalid code violation.

Other errors might transform a valid code word into another different valid code word. This type of error cannot be detected by the decode logic when they occur. However, a single-bit error of this type always changes the Running Disparity value. The Running Disparity rule will detect these errors, immediately or some time after the error had occurred.

Example 1:

(-1) 101010 1010 (-1) was transmitted
(-1) 100010 1010 (-3) was received

The transformation from one code word (101010) to another valid code word (100010) will be detected as an error because when 100010 is received with previous RD=-1 will result in an illegal RD value of -3 afterward.

Example 2:

(-1) 101010 1010 (-1) 101010 1010 (-1) 101010 1010 (-1) was transmitted
(-1) 101110 1010 (+1) 101010 1010 (+1) 101010 1010 (+1) was received

The transformation from one code word (101010) to another valid code word (101110) will not be detected as an error because all the subsequent data bytes are zero-disparity and do not provide checking against Running Disparity rule. The error detection will not occur until a non-zero disparity code word is received.

In order to localize such behavior, it is necessary to packetize data in some fashion. The special characters should be used as packet delimiters because they are non-zero disparity code words. They will terminate the propagation of an erroneous RD value and flag a previous receiving error.

Obviously, errors which are double-bit or more, and transform one code word into another valid code word without altering the RD value, cannot be detected by the 8B10B native decode logic. Such errors can be shown as:

Example 3:

(-1) 101010 1010 (-1) was transmitted
(-1) 100110 1010 (-1) was received

The transformation from one code word (101010) to another valid code word (100110) cannot be detected. Higher level error protection mechanisms such as a cyclic-redundancy-check are needed.

Another issue that warrants more discussion is the effect of transmission errors on the user data. For example,

(+1) 011000 1010 (-1) was transmitted
(+1) 011001 1010 (+1) was received as the result of a single-bit transmission error.

The transmitted user data was 00000 101 and the received user (decoded) data is 00110 101. Observation shows that there is a double-bit error in the user data. Therefore, a simple user parity scheme (horizontal parity) will not protect user data from single-bit error in the encoded data transmission.

D.1.8 Summary:

The ANSI Fibre Channel 8B10B code is an excellent block code for serial data transmission. It generates DC-free signals through the control of running disparity. It has complete coverage and indication for any single-bit error event.

D.1.9 References:

1. *US patent 4,486,739; Dec 4, 1984*
   Peter A. Franaszek, Albert X. Widmer; IBM
2. *US patent 4,665,517; May 12, 1987*
   Albert X Widmer; IBM
3. *A DC Balanced, Partitioned Block, 8B10B Transmission Code*
   A.X. Widmer, P.A. Franaszek
   IBM Journal of Research and Development, Vol. 27, No 5, September 1983

D.2 10B12B Transmission Block Code

The AMD proprietary 10B12B coding scheme is based on the same disparity, RD, error detection, principles and 5B6B coding table as the 8B10B coding scheme discussed in the previous section. In 10B12B coding, when two 5B6B subblocks are concatenated according to the RD rule to maintain Running Disparity at +1 or -1 values, a 10B12B block code is constructed. The coding scheme has its own unique special character set listed in the TAXI-275 data sheet and below in Table D.3. The 10B12B code is an obvious extension of the 8B10B code without the encoding exceptions.

Table D.3. 10B12B Encode/Decode Table

5B6B Subblock Table (left-most encoded bit is transmitted first serially)

5-Bit Data	Encoded	(Disparity)	Alternate	(Disparity)
00000	011000	(-2)	100111	(+2)
00001	100010	(-2)	011101	(+2)
00010	010010	(-2)	101101	(+2)
00011	110001	(0)	none	
00100	001010	(-2)	110101	(+2)
00101	101001	(0)	none	
00110	011001	(0)	none	
00111	111000	(0)	000111	(0)
01000	000110	(-2)	111001	(+2)
01001	100101	(0)	none	
01010	010101	(0)	none	
01011	110100	(0)	none	
01100	001101	(0)	none	
01101	101100	(0)	none	
01110	011100	(0)	none	
01111	101000	(-2)	010111	(+2)
10000	011011	(+2)	100100	(-2)
10001	100011	(0)	none	
10010	010011	(0)	none	
10011	110010	(0)	none	
10100	001011	(0)	none	
10101	101010	(0)	none	
10110	011010	(0)	none	
10111	111010	(+2)	000101	(-2)
11000	001100	(-2)	110011	(+2)
11001	100110	(0)	none	
11010	010110	(0)	none	
11011	110110	(+2)	001001	(-2)
11100	001110	(0)	none	
11101	101110	(+2)	010001	(-2)
11110	011110	(+2)	100001	(-2)
11111	101011	(+2)	010100	(-2)

D.2.1 An Example of 10B12B Encoding

The easiest way to understand the 10B12B code is to follow an example.

Let there be four bytes of user data to be encoded.

```
HEX user data:            272            308              2FF             10A
bit definitions:     KJHGF EDCBA    KJHGF EDCBA     KJHGF EDCBA     KJHGF EDCBA
binary user data:     10011 10010    11000 01000     10111 11111     01000 01010
bit definitions:      abcde fghjk    abcde fghjk     abcde fghjk     abcde fghjk
10B12B subblocks:     01001 11001    00010 00011     11111 11101     01010 00010
10B12B notation:       D18 . D19      D8 . D24        D31 . D23       D10 . D8
initial RD:            (-1)
5B6B encode; (RD)      010011     (-1)
5B6B encode; (RD)          110010     (-1)
5B6B encode; (RD)              111001     (+1)
5B6B encode;  (RD)                     001100     (-1)
5B6B encode;  (RD)                      101011     (+1)
5B6B encode;  (RD)                        000101     (-1)
5B6B encode;  (RD)                              010101     (-1)
5B6B encode; (RD)                                  111001     (+1)

final RD:              (+1)

serial bit stream:     010011110010111001001100101011000101010101111001
```

Observations: within the 48-bit stream, there are 25 ones and 23 zeros. The value of Running Disparity changes from an initial -1 to a final +1, with intermediate values at subblock boundary of either -1 or +1. For each subblock, the encoding decision is made based upon the Running Disparity rule.

D.2.2 Special Characters

When two 5B6B subblocks are concatenated according to the RD rule, the resultant 12-bit code word is mapped into the $2^{12} = 4096$ code space. There are some code words which satisfy the run-length and transition density requirements although the code words are not used by legally encoded data. These code words (special characters) are available for representation of other data transmission functions, such as block alignment. They are:

Special Code Name	Input	Current RD–		Current RD+	
		abcdei	fghj	abcdei	fghhj
K28.D16	00	001111	100100	110000	011011
K28.D13/D18	01	001111	101100	110000	010011
K28.D14/D17	10	001111	011100	110000	100011
K28.D13+	11	001111	101100	no alternate	

All special character transmissions must follow the same Running Disparity rule.

The special characters K28.D13 and K28.D13+ have 5 consecutive ones following 2 zeros or 5 consecutive zeros following 2 ones. These combinations do not occur in a normally encoded data stream. These two special characters can be used for byte synchronization--the unique bit pattern is used to identify the byte boundary for serial-to-parallel conversion.

APPENDIX

E INTRODUCTION TO HIGH SPEED COMMUNICATION STANDARDS

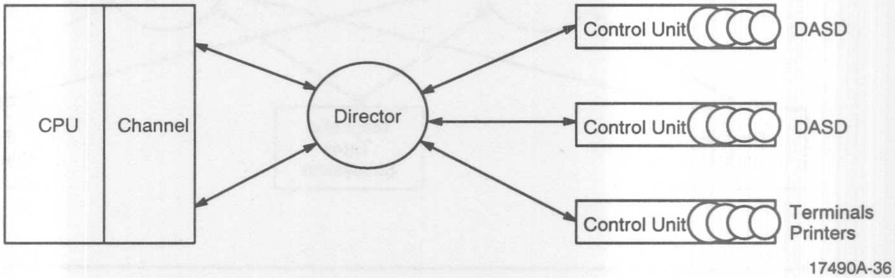
E.1 INTRODUCTION TO ESCON

IBM announced the Enterprise System Architecture ESA/390 in 1990 to meet the increasing customer demands and challenges of its mainstream business. ESCON is an integral part of the system in terms of providing better performance and improved configuration flexibility through an innovative new architecture for I/O and system connectivity. ESCON utilizes the newest technologies in fiber optic communication to realize 200 Mbaud links. The essence of ESCON lies with two major components: the point-to-point protocol and the Director. The "switched point-to-point" interconnection scheme uses "Directors" to dynamically switch multiple point-to-point links, thereby achieving total system connectivity and throughput (see Figure E.1).

The new architecture, topology and technology bring together:

- configuration management under program control,
- high availability, reliability,
- simplified device connectivity and sharing,
- increased connectivity distances,
- a basis for growth,
- improved I/O performance and efficient use of channels,
- reduced cable bulk and weight.

Figure E.1 ESCON Channel Switched Point-to-Point Topology



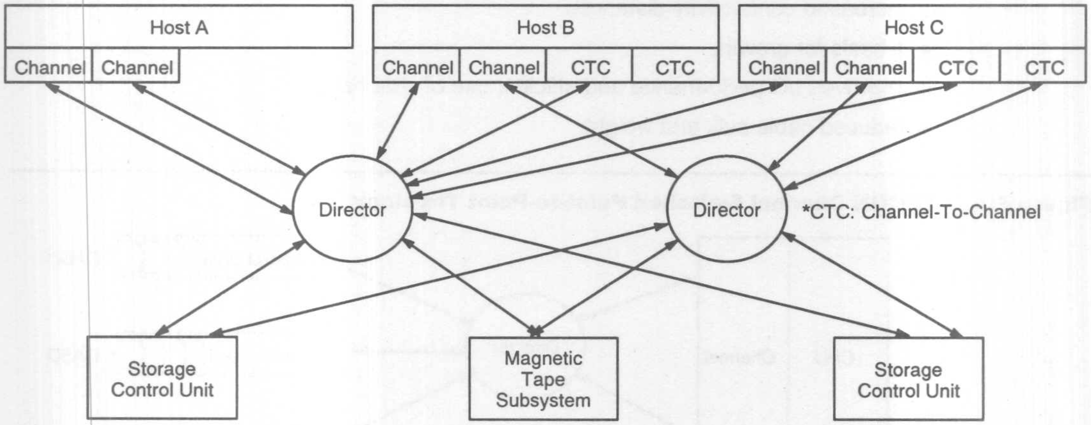
E.1.1 The Old Copper Channel versus ESCON:

A "Channel" is what an IBM mainframe computer uses to communicate with its peripheral devices such as DASD (Direct Access Storage Device), magnetic tape controllers, terminal controllers, printer controllers, etc. The "bus-tag" channel connection (OEMI channel) is a daisy-chained scheme with a total distance limit of 400 feet and maximum throughput of 4.5 MByte/s. The number of channels required to support large system images can become very large and difficult to manage, while at the same time typical channels are underutilized (less than 50% active). Connectivity, as well as sharing data across multiple systems through daisy-chained scheme is increasingly complex and expensive. As system requirements and performance grew over time, the limitations of the parallel channel had been reached.

The ESCON Channel employs new protocols and topology for information exchange. The number of physical connections is no longer dictated by logical connectivity needs. Fiber optic cable eliminates many of the undesirable electrical properties of copper cables while at the same time is much easier to handle due to the reduced weight/bulk (at a fraction of the “blue” cable) and better flexibility. The point-to-point distance is extended to 3 km at the potential maximum link band width of 20 MByte/s. Daisy-chain is no longer used and the system/devices are configured around the Directors. Two ESCON Directors are allowed within an end-to-end connection thus providing a maximum distance of 9 km (at least one Director in a two-Directors path must be static). The Directors are responsible for setting up logical paths between channels and control units and make simultaneous communication of multiple paths possible. If reliability or availability is a concern, Directors provide redundant logical paths and physical connections when multiple links are installed.

ESCON protects user software investment since it is compatible at the application program level. With the installation of an “ESCON Converter” (a box that converts bus-tag channel signal to/from ESCON channel signals), a mixed environment of System/370 channels and ES/390 ESCON channels will co-exist and therefore ease the migration to ESCON.

Figure E.2 ESCON Channel Dynamic Connectivity without Single-Point-of-Failure



17490A-37

The benefits of ESCON can be summarized as:

- Less disruptive installation and reconfiguration since processors and control units may be added or removed while the system is in operation, eliminating many planned down times.
- Increased channel-to-device connection distances. ESCON allows I/O devices to be located up to 9 km from the processor, enabling campus-wide configuration, disaster recovery, and remote I/O pools.

- Dynamic switching architecture improves redundancy and availability of connectivity at the same or lower level of cabling complexity.
- New I/O subsystem capabilities such as integrated channel-to-channel functions, increase data rate.
- Campus-wide I/O sharing with reduced complexity through Directors.
- Centralized configuration management. When ESCON Directors are used, a new host program allows entire campus configuration to be managed from a single point of control. Error information may be centralized and automatically analyzed.
- Simplified interconnection by the usage of light weight fiber optic cable. It also provides device location flexibility, reduces the number of physical connections, cable bulk, and under-floor disruption; all at reduced cabling cost. A typical ESCON installation requires fewer channels and far less cabling than a bus-tag channel installation with the same level of redundancy and connectivity.

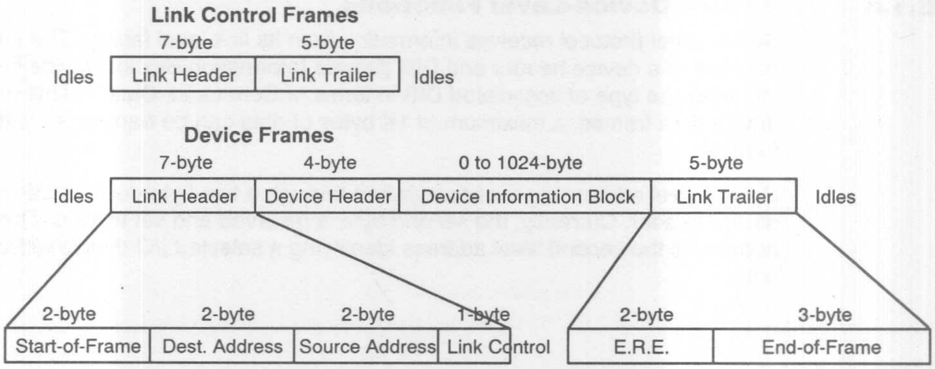
E.1.2 ESCON Physical Links

The ESCON channel supports 62.5/125 and 50/125 (core diameter/cladding diameter) multi-mode (graded index) glass fibers. Distances of 9 kilometers (3km x 3) can be reached using 62.5/125 fiber, while 50/125 fiber supports maximum distance of 6 km (2 km x 3). Currently, Siemens and AT&T are the approved manufacturers of ESCON optical transceivers. The optical components include 1300 nanometer wavelength LED sources and drivers, PIN diode photo detectors and amplifiers, as well as duplex connectors. Patch panels, connectors, and splices are also considered part of the physical links.

E.1.3 ESCON Link-Level Functions

Link-Level protocol is responsible for transmission and reception of frames through physical connections. It manages the link using the exchange of link control frames, performs link address management and recovers from link-level error conditions. Two Link-Level facilities are connected together via a physical path which includes fiber optic cable and components.

Figure E.3 ESCON Channel Link Level Frame Format and Header



17490A-38

Link control frames generally occur in request-response pairs; such as ELP-LPE (Establish Logical Path–Logical Path Established), RLP-LPR (Remove Logical Path–Logical Path Removed) pairs. ACK (Acknowledgment), LRJ (Link-level Reject), and LBY (Link-level Busy) are also link control frames. The types of link control frames are encoded in the Link Control byte within the Link Header.

Another important link control protocol function is address acquisition. When an attachment is initialized, it needs to exchange ALA-ACK (Acquire Link Address-Acknowledgment) with the Director port to which it is attached in order to obtain its assigned link address.

Primitive level signaling (sequencing) is also part of the link-level protocol. This includes the exchange of sequences such as NOS (Not-Operational), UD (Unconditional-disconnect), UDR (Unconditional-disconnect-response), OLS (Offline) and Idle. This function is performed primarily as part of error recovery procedure or prior to link initialization.

A sequence is the consecutive occurrence of the same Ordered Set symbols. The ESCON Ordered Sets are defined by Table E.1. (For the actual symbol bit pattern of the Ordered Set, please refer to the 8B10B section.)

Table E.1 ESCON Ordered Sets for Sequence Functions

Sequence Function	Ordered Set
Not-Operational (NOS)	K28.5 D0.2
Unconditional-Disconnect (UD)	K28.5 D15.2
Unconditional-Disconnect-Response (UDR)	K28.5 D16.2
Off-line (OLS)	K28.5 D24.2

Other link-level functions include: data integrity check through a 16-bit cyclic-redundancy-check (CRC), link error detection and recovery, address validation and management.

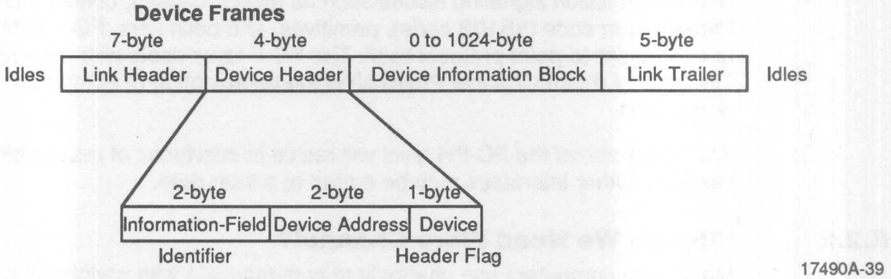
When link-level protocol is completed, each frame is stripped of its link-level header and trailer. The information field is further processed by the device-level protocol.

E.1.4 ESCON Device-Level Functions

Device-level protocol receives information from its link-level facility. The information consists of a device header and DIB (Device Information Block). Device header identifies the type of associated DIB in terms of Command, Data, Status or Control. For device data frames, a maximum of 1K bytes of data can be transferred within each frame.

Device-level addressing is accomplished through a two-byte device address within the device header. Currently, the second byte is reserved and set to zero. The device address is the second level address identifying a selected I/O device within a control unit.

Figure E.4 ESCON Channel Device Frame Format and Header



Device-level protocol is also responsible for maintaining pacing parameters for speed matching and management of logical path information.

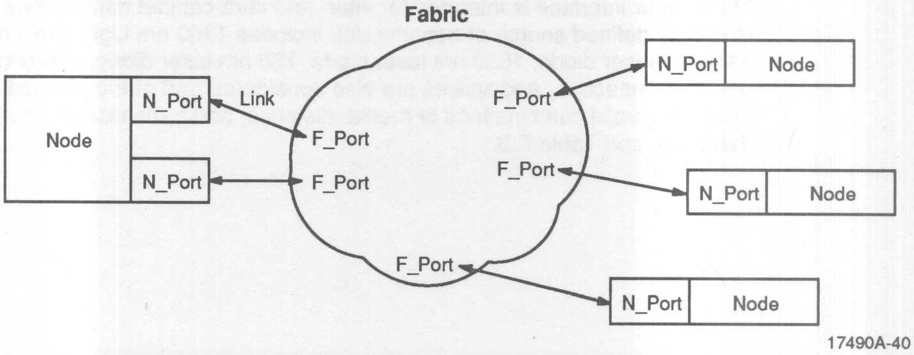
E.2 ANSI FIBRE CHANNEL STANDARD (FCS)

E.2.1 Introduction to FCS

ANSI X3T9.3, the same group that defined the standards of Intelligent Peripheral Interface (IPI) and High Performance Parallel Interface (HIPPI), has been working on the Fibre Channel standard to define the future channel architecture and connectivity scheme. Many organizations participated in the efforts along the way and contributed to the release of a working document FC-PH Rev 2.10 for review and comment in March of 1991. Members of the working group include computer system (mainframe, work-station, desk top machines) manufacturers, research institutes, network system manufacturers, peripheral manufacturers, cable and connector manufacturers, and silicon and optical component vendors.

The essence of Fibre Channel lies with two major components: the point-to-point protocol and the Fabric. The "switched point-to-point" interconnection scheme uses "Fabric" to dynamically switch multiple point-to-point links, thereby achieving total system connectivity. Graphically illustrated below, each "Node" can be a mainframe processing unit, Direct Access Storage Device, distributed processing unit, or any other peripheral unit.

Figure E.5 Fibre Channel Terminology and Switched Point-to-Point Topology



The Fibre Channel document is divided into several pieces. FC-0 is the level dealing with transmission signaling issues such as media, cabling, connectors. FC-1 defines transmission code (8B10B code), primitives, and delimiters. FC-2 defines frame format and the point-to-point protocol itself. The FC-F layer deals with the Fabric interfaces. The FC-PH (Fibre Channel Physical) is the combination of FC-0, FC-1 and FC-2 documents.

The levels above the FC-PH level will reside in interfaces of many different kinds of services. Other interfaces may be added at a later date.

E.2.2 Why Do We Need Fibre Channel?

Mainframe computers use channels to communicate with peripheral devices such as disk storage, tape storage, terminal controllers, printer controllers, etc. The traditional channel connection is done through daisy-chained bus scheme. Connectivity, as well as sharing data across multiple systems through a daisy-chained scheme is increasingly complex and expensive.

Small systems use SCSI (Small Computer System Interface) which has grown over the years from a few kbyte/s of throughput into today's SCSI-2 capable of handling more than 20 Mbytes/s. SCSI also uses a parallel chained bus in a limited distance environment. The recent development of numerous termination techniques for SCSI cabling systems indicates the challenges involved when faster signals start to propagate down the parallel cables.

As system requirements of performance grew over time, the limitations of the parallel channel have been reached. The Fibre Channel utilizes new protocols and topology for information exchange. The number of physical connections is no longer dictated by logical connectivity needs. High speed serial transmission media has replaced the parallel bus. Fiber optic cable eliminates many of the undesirable electrical properties of copper cables while at the same time is much easier to handle due to the reduced weight/bulk and better flexibility. The point-to-point distance is extended to a few kilometers.

E.2.3 FCS Physical Links

The Fibre Channel FC-0 layer supports multiple transmission media which includes 62.5/125 and 50/125 (core diameter/cladding diameter) multi-mode (graded index) glass fibers, 9/50 single-mode glass fiber, 75-Ω CATV, miniature coaxial cables, and STP Type 1. The single-mode fiber is used for distance as long as 50 kilometers in metropolitan connections. Multi-mode fiber allows distances up to 3 km for campus environment. The copper interface is intended for inter- and intra-cabinet connections up to 50 m. Currently defined source of transmission includes 1300 nm Light-Emitting-Diode (LED), 1340 nm laser diode, 1500 nm laser diode, 780 nm laser diode, and cable driver. Patch panels, connectors, and splices are also considered part of the physical links defined by FC-0. The valid combinations of media, distance, and transmission source are listed in Table E.2 and Table E.3.

Table E.2 Fibre Channel FC-0 Valid Combinations

Baud Rate	Valid Cable Plant
1062.5	100-SM-LL-L, 100-SM-LL-I, 100-TV-EL-S, 100-MI-EL-S
531.25	50-SM-LL-L, 50-M5-SL-I, 50-TV-EL-S, 50-MI-EL-S
265.625	25-SM-LL-L, 25-SM-LL-I, 25-M5-SL-I, 25-M6-LE-I, 25-TV-EL-S, 25-MI-EL-S, 25-TP-EL-S
132.8125	12-M6-LE-I, 12-TV-EL-S, 12-MI-EL-S, 12-TP-EL-S

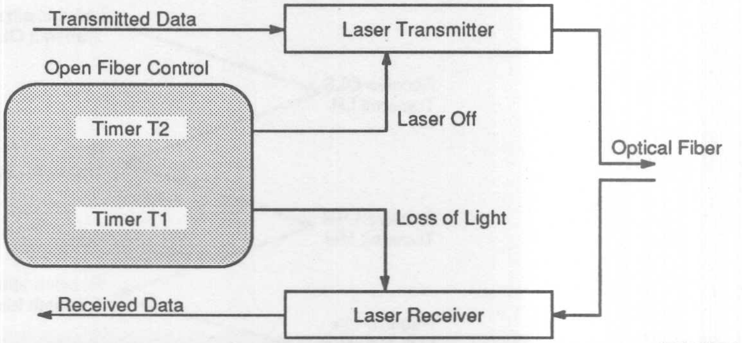
Table E.3 Nomenclature of FC-0 (Cable Plant Speed-Media-Transmitter-Distance)

Speed:	100 = 100 Mbyte/s or 1062.5 Mbaud
	50 = 50 Mbyte/s or 531.25 Mbaud
	25 = 25 Mbyte/s or 265.625 Mbaud
	12 = 12.5 Mbyte/s or 132.8125 Mbaud
Media:	SM = single-mode fiber
	M5 = multi-mode fiber (50 µm)
	M6 = multi-mode fiber (62.5 µm)
	TV = video cable
	MI = miniature cable
	TP = twisted pair cable
Transmitter:	LL = long wavelength laser (1300 nm)
	SL = short wavelength laser (780 nm)
	LE = long wavelength LED (1300 nm)
	EL = electrical
Distance:	L = long distance (up to 10 km)
	I = intermediate distance (up to 2 km)
	S = short distance (up to 10 or 50 m)

E.2.3.1 Laser Safety: Open Fiber Control (OFC)

This physical level protocol is a safety interlock mechanism whenever laser (LL, SL) is used as the optical source. When optical fiber is disconnected or cut, the OFC guarantees that the human exposure to laser emission is under a certain level. The following diagram helps to explain how OFC functions.

Figure E.6 Fibre Channel FC-0 Open Fiber Control



17490A-41

When the laser receiver asserts Loss of Light (LOL) to indicate an open fiber condition, the OFC will turn off the transmitting laser immediately and start timer T1. At the opposite end of this link, LOL will be asserted and its laser turned off. When T1 expires, OFC turns on the laser to transmit for a duration controlled by timer T2. During T2, if the receiver detects incoming light, then the link is assumed closed and will go through proper recovery procedure. If LOL persists throughout T2, then T1 will be started again and the laser shut off. In an open fiber situation, the average laser power emitted is proportional to the ratio of T2/T1, which is set to meet the requirement of laser safety standard.

E.2.3.2 FCS Transmission Block Code and FC-1

The Fibre Channel FC-1 layer defines the block code in data transmission. The code chosen is an 8B10B code developed by IBM and implemented in ESCON. Also defined by FC-1 are the Start Of Frame (SOF) delimiters, End Of Frame (EOF) delimiters, Idle and other primitives for low-level signaling.

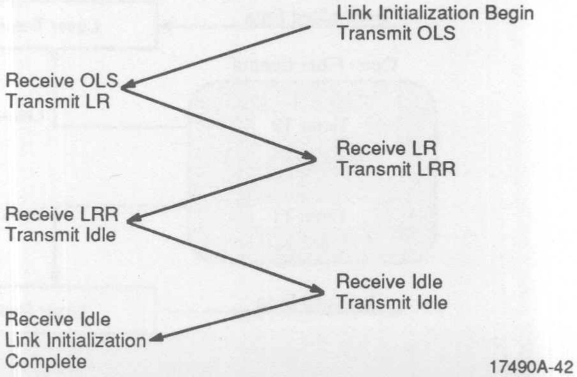
A Sequence is the consecutive occurrence of the same Ordered Set symbols. The defined Primitive Sequences are Not-Operational (NOS), Off-line (OLS), Link Reset (LR), Link Reset Response (LRR), and Idle (Idle), as listed in Table E.4. (For the actually symbol bit pattern of the Ordered Set, please refer to the 8B10B section.)

Table E.4 Fibre Channel Ordered Sets for Sequence Functions

Sequence Function	Ordered Set			
Off-line(OLS)	K28.5	D21.1	D10.4	D21.2
Not-Operational (NOS)	K28.5	D21.2	D31.5	D5.2
Link_Reset (LR)	K28.5	D9.2	D31.5	D9.2
Link_Reset_Response (LRR)	K28.5	D21.1	D31.5	D9.2
IDLE	K28.5	D21.4	D21.5	D21.5

Primitive Sequence protocol provides a mechanism for link initialization or recovery from link failure. For example, when a port has been powered-up or reset, the protocol for link initialization is performed by the sequence of transmit_OLS, receive_LR, transmit_LRR, receive_Idle, transmit_Idle. Other protocols using Primitive Sequences are On-line-to-Off-line protocol, Link Failure protocol, and Link Recovery protocol.

Figure E.7 Fibre Channel Primitive Sequence Protocol Example (Link Initialization)



E.2.4 Fibre Channel Frame Format

The Fibre Channel frame in Figure C.8 is the integral unit of data being exchanged. Frames are separated by Idle words (each Idle word is 4 bytes long). A minimum of two Idles are required to precede a frame at the receiving port. The first word of the frame is the Start Of Frame (SOF) delimiter and the last word is the End Of Frame (EOF) delimiter. Between the SOF and EOF is the content of the frame, which contains a 24-byte Frame Header, 0 to 2112 bytes of data, and a 4-byte Cyclic Redundancy Check (CRC-32) checksum. At the beginning of the data field, there can be 4 types of optional headers for any higher level protocol function.

Figure E.8 Fibre Channel FC-2 Frame Format

Idles	SOF	Frame Header	Data Field	CRC	EOF	Idles
	4-Byte	24-Byte	0 to 2112-Byte	4-Byte	4-Byte	

17490A-43

The Frame Header in Figure C.9 contains R_CTL (Routing Control, 1-byte), D_ID (Destination Identifier, 3-byte), S_ID (Source Identifier, 3-byte), TYPE (data structure Type, 1-byte), F_CTL (Frame Control, 3-byte), SEQ_ID (Sequence Identifier, 1-byte), DF_CTL (Data Field Control, 1-byte), SEQ_CNT (Sequence Count, 2-byte), OX_ID (Originator Exchange Identifier, 2-byte), RX_ID (Responder Exchange Identifier, 2-byte), and Parameter (4-byte) fields.

Figure E.9 Fibre Channel FC-2 Frame Header

R_CTL	D_ID	
Reserved	S_ID	
TYPE	F_CTL	
SEQ_ID	DF_CTL	SEQ_CNT
OX_ID		RX_ID
Parameter		

17490A-44

There are two categories of frames: Data (Link_Data, Device_Data) frames and Link_Control (Link_Continue, Link_Response) frames. Data frames are used to transfer information such as data, control, header, and status from one N_Port to a destination N_Port. Every Data frame has a set of Link_Control responses and a R_RDY primitive associated with it. Link_Control frames are link level frames (do not propagate to higher levels) corresponding to the delivery of each Data frame. A Link_Continue (ACK_1 or ACK_N) frame indicates successful delivery of a Data frame and Link_Response (BSY or RJT) frame indicates unsuccessful delivery of a Data frame.

E.2.5 FCS Link-Level Functions

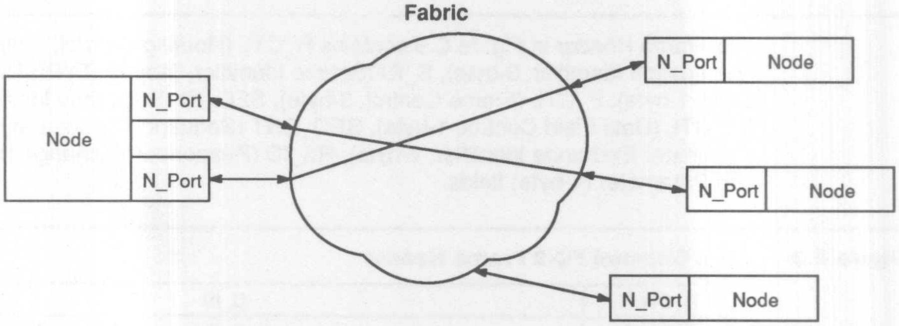
Link-Level protocol is responsible for transmission and reception of frames through physical connections. It manages the link using the exchange of link control frames, performs link address management and recovers from link-level error conditions. Two Link-Level facilities are connected together via a physical path which includes cable plant and transceiver components.

Each N_Port logically performs point-to-point communication with another N_Port at any given time. The FCS ports can operate in one of three modes: half-duplex, duplex, and dual-simplex.

FC-2 also defines three classes of services. See Figure E.10. For Class 1 service (Dedicated Connection), the Fabric guarantees full bandwidth between two communicating N_Ports after the Connection is properly established. Class 2 service (Multiplex) allows frames to be multiplexed at frame boundary with notification of non-delivery of frame. Class 3 is a connectionless service with best-effort delivery of frames by the fabric, without any notification of delivery (ACK) or non-delivery (BSY or RJT), sometimes also referred to as the "datagram" service.

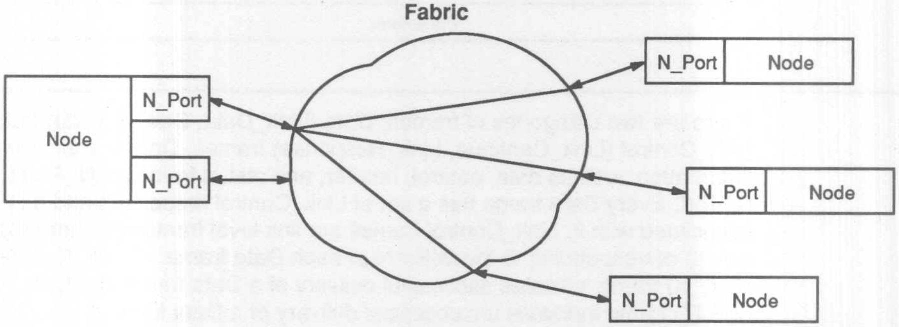
Figure E.10 Fibre Channel Physical Link Layers Class1 and Class 2

Class 1 (Dedicated Connection) Illustrated



17490A-45

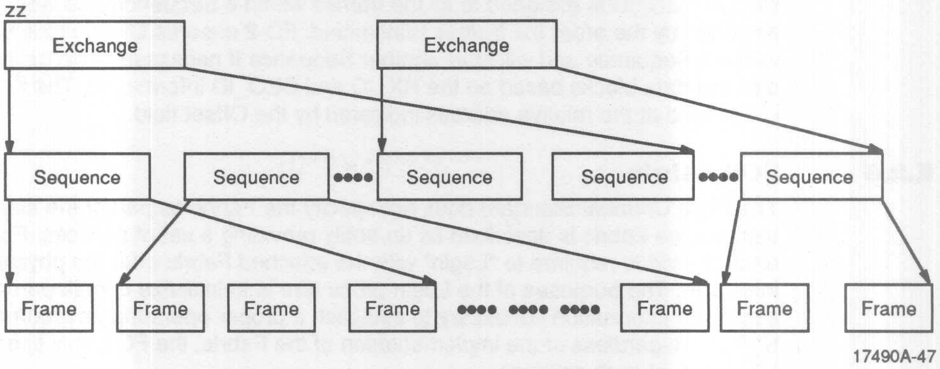
Class 2 (Multiplex) Illustrated



17490A-46

Two communicating N_Ports use a hierarchy of building blocks for data movement and control. The smallest entity that can be dealt with is a frame. A Data Frame carries higher level data within its payload and a Link Control Frame is used by Fibre Channel protocol to exchange link-level information and maintain link-level control functions. The ACK frame is an example of a Link Control frame.

Figure E.11 Fibre Channel Building Block Hierarchy



One or more related Data Frames can be transmitted unidirectionally and the corresponding Link Control Frame in response comprise a Sequence. See Figure C.11. Each Sequence is assigned a Sequence Identifier (SEQ_ID) so that an N_Port can collect the related frames. A Sequence is started by a frame with Start_Of_Frame_initiate (SOFi) and terminated by a frame with End_Of_Frame_terminate (EOFt). Each frame within a Sequence carries a Sequence Count (SEQ_CNT), which is incremented with each frame transmitted.

One or more non-concurrent, related Sequences make up an Exchange. An Exchange may contain either unidirectional or bidirectional data flow. The initiator of an Exchange assigns an Originator Exchange identifier (OX_ID) and the responder assign its unique Responder Exchange identifier (RX_ID). Only one Sequence is allowed to be active at any given time within any single Exchange. The combination of S_ID, D_ID, SEQ_ID and SEQ_CNT uniquely identify any frame within an Exchange. An Exchange is the basis for data transfer between two N_Ports. When an Exchange is initiated and accepted, there is a binding of resources in both the Originator and the Responder for the duration of the Exchange.

FC-2 manages activation and deactivation of Exchanges, initiation and termination of Sequences, assignments of OX_ID and RX_ID, transfer of Sequence initiatives, assignments of SEQ_ID, segmentation and reassembly of data according to Exchange/Sequence/Frame, Sequences and counting of frames. In addition, for Class 1 and 2 services, FC-2 also manages any frame Sequence error detection and notification.

Flow Control is the FC_2 control process to pace the flow of frames to prevent overrun at the receiving end of a link. It is managed using Credit, Credit Count, ACK_1, ACK_N, receiver ready (R_RDY) along with other frames. Class 1 service uses end-to-end flow control (between two N_Ports). Class 2 service uses end-to-end and buffer-to-buffer flow control. Class 3 service uses only buffer-to-buffer flow control (between N_Port and F_Port). Credit is the number of receive buffers allocated and available to a transmitting port. The running count of the Credit is the Credit_CNT. Credit_CNT is incremented with each frame transmitted and decremented with each ACK_1 (decremented by n if ACK_N is received) received from the destination. The Sequence initiator N_Port ensures the Credit_CNT does not exceed Credit. End-to-end flow control is managed with N_Port Credit_CNT, and buffer-to-buffer flow control is managed with Fabric Credit_CNT.

FC-2 performs segmentation and reassembly of blocks of data using Exchange/Sequence/Relative Offset. During Segmentation process, FC-2 first determines the size of payload for each data frame and computes the Offset of the payload in each frame. Unique SEQ_ID is assigned to all the frames within a Sequence and SEQ_CNT is assigned by the order the frames transmitted. FC-2 ensures Offset value will not wrap within a Sequence and will start another Sequence if necessary. The recipient reassembles the data blocks based on the RX_ID and SEQ_ID information. The first byte of each is received at the relative address indicated by the Offset field.

E.2.6 FCS Fabric

The Fibre Channel Standard does not specify the Fabric as part of the standard. Instead, the Fabric is described as an entity providing a set of services. For example, each N_Port is required to "Login" with the attached Fabric after the physical link is initialized. The purposes of the Login procedure is to initialize certain parameters and exchange information necessary to establish a proper operating environment of the N_Port. Regardless of the implementation of the Fabric, the FCS only specifies the interfaces of such services.

The freedom of implementation choice is therefore given to the user/implementor. It is anticipated that a variety of Fabric architectures will arise. Each FCS environment will be able to establish its own cost-versus-performance tradeoff, based upon its unique set of criteria.

Among many possible Fabric architectures, cost and performance will vary greatly. A simple and low cost fabric may only employ a broadcast scheme, while an expensive and high performance fabric may contain multiple redundant cross-point switches to enhance throughput. All of them will appear to have the same interface at the F_Port and provide the same services to all the attached N_Port.

E.3 ESCON versus Fibre Channel

A commonly asked question after someone is introduced to ESCON and Fibre Channel Standard is always a comparison or compatibility between the two.

IBM has made a great deal of contribution within the ANSI working group, drawing from their ESCON experiences. There are certain resemblances between the two at first glance. Upon further study, one would realize that the FCS is a much more encompassing standard and ESCON would only constitute a subset of FCS. Specifically, ESCON protocol only coincides with the Class-1 (Dedicated Connection) protocol in FCS. Furthermore, FCS has different frame formats, different frame headers and sizes, and different CRC, which makes direct (plug) compatibility impossible.

In addition, FCS is structured such that a variety of architectures can be used for the Fabric, be it cross-point switch or store-and-forward switch or even distributed switching. ESCON Directors employ a dedicated blocking switching scheme which supports its link layer protocol.